Three years of graphs and music : some results in graph theory and its applications

Nathann Cohen

To cite this version:

Nathann Cohen. Three years of graphs and music : some results in graph theory and its applications. Discrete Mathematics [cs.DM]. Université Nice Sophia Antipolis, 2011. English. NNT : . tel-00645151

UNIVERSITÉ DE NICE-SOPHIA ANTIPOLIS - UFR SCIENCES

ÉCOLE DOCTORALE STIC
SCIENCES ET TECHNOLOGIES DE L'INFORMATION
ET DE LA COMMUNICATION

T H È S E

pour obtenir le titre de

Docteur en Sciences

de l'Université de Nice - Sophia Antipolis
Mention : INFORMATIQUE

Présentée et soutenue par

Nathann COHEN

Three years of graphs and music

Some results in graph theory and its applications

Thèse dirigée par **Frédéric HAVET**

préparée dans le Projet MASCOTTE, I3S (CNRS/UNS)-INRIA

soutenue le 20 octobre 2011

Rapporteurs :	Jørgen BANG-JENSEN	-	Professor University of Southern Denmark, Odense
	Daniel KRÁL'	-	Associate Professor, Charles University, Prague
	András SEBŐ	-	Directeur de Recherches, G-SCOP Laboratory, Grenoble
Directeur :	Frédéric HAVET	-	Chargé de Recherches, Mascotte, Sophia-Antipolis
Examinateurs :	Stéphane BESSY	-	Maitre de Conférences, LIRMM, Montpellier
	Victor CHEPOI	-	Professeur, Université de la Mediterranée, Luminy
	Ioan TODINCA	-	Professeur, LIFO, Université d'Orléans
Invité :	Stéphane PERENNES	-	Directeur de Recherches, Mascotte, Sophia-Antipolis

Three years of graphs and music

Some results in graph theory and its applications

Nathann Cohen

20 october 2011

Contents

A few words

This thesis is about three years happily spent playing with graphs, walking absentmindedly in corridors or standing in front a whiteboard, while whistling or listening to music. It contains results obtained with several coauthors, as well as many of the elegant and more classical results I met reading books or papers, which to my sense gave a wider understanding of the problems presented. It has for background the laboratory of the MASCOTTE team set up by Jean-Claude Bermond and now headed by David Coudert, its offices whose doors open on whiteboards covered with neat and well-thought formulas – or hasty drawings depending on the character of the inhabitants – and its collection of open problems on which all of us have once tried our luck.

I owe the time I spent on graph coloring to my advisor Frédéric Havet, on problems of hypergraphs to Jean-Claude Bermond, and the mixture of flows, graph decompositions and linear programming that helped me all along is definitely MASCOTTE's own tune.

I have met in Sophia-Antipolis and elsewhere quantity of people who never hesitated to sit down and think whenever a question was asked, for the sheer pleasure of doing so. I thank them all wholeheartedly, and in particular Stephane Perennes who, unbeknownst to him, probably taught me all I know of graphs today by mixing together in the same sentences the most remote notions, legitimately "confusing" one word with another, leaving to me to notice days later that "indeed, in this context replacing the natural notion with this apprently unrelated one was the best way to look at the problem at hand".

Because I regularly contributed to the math software Sage [145] during the last two years, I also found a place inside of a much wider family of mathematicians, sticking together for their interest in translating into algorithms the words of mathematical theory. It probably would have taken me much longer, if not for them, to notice how deeply mathematics is an experimental science, and I can but wonder at how much I would have missed if not for those nights spent implementing, fixing, testing and rewriting many of the most elementary algorithms graph theory has to offer, and that I wrongly assumed I had understood before explaining them to a computer. My fellow Sage developpers I want to thank for their energy, for their infinite enthusiasm, for the discussions at any time of day or night thanks the timezones they cover, and for constantly daring me to work on the whole spectrum of mathematics at the same time.

Hence, I am largely indebted to my coauthors from here and abroad, and to all of the genuinely curious people I have had the pleasure to work with along the years. Let me also thank Jørgen Bang-Jensen, Daniel Kràl and András Sebő who kindly accepted to read this manuscript as well as Stéphane Bessy, Victor Chepoi, and Ioan Todinca who were in my jury.

Oh. And because I never miss an occasion to advertise good books – especially when it is out of place – let me mention many names that were always around during those three years of graphs and music : Alessandro Barricco, Carlos Castaneda, Michel Foucault, Allen Ginsberg, Hermann

Hesse, Jack Kerouac, Milan Kundera, Robert Pirsig, Henry-David Thoreau (Walden), Oscar Wilde, and Stefan Zweig. Just yesterday I finished "The Island" from Aldous Huxley, which was a fantasic read too.

I can find no words for my mother, who by now is aware that the largest part of everything I do she taught me herself.

Nathann

Pretty conjectures

On this page are presented four conjectures which, along with many unsolved others represent the largest part of the time I spent working during those three years of Ph.D.

Though these do not appear in the following pages, open questions are in many respects much more interesting that known results. As they taught me much and are, after all, probably the best of what I learned during all these years, it would have been terribly ungrateful not to give them some space in this document.

Chvátal's conjecture

This conjecture first appeared in 1974 in an article from Vašek Chvátal [48].

Given a hereditary Set System, i.e. a family of sets $\mathcal{F} \subseteq 2^X$ such that $\forall S \in \mathcal{F}$ any subset of S is also an element of $\mathcal{F}$, one obtains an *intersecting family* (i.e. a family $\mathcal{I} \subseteq \mathcal{F}$ such that any two elements of $\mathcal{I}$ intersect) by picking an element $x \in X$ and considering the family $\mathcal{F}_x$ of all the elements of $\mathcal{F}$ containing x.

Is $\max\limits_{x \in X} |\mathcal{F}_x|$ the maximum size of an intesecting family in $\mathcal{F}$?

The Cacetta-Häggkvist conjecture

This conjecture is from 1978, and first appears in [42] from Caccetta and Häggkvist.

Given a directed graph D on n vertices such that any vertex of D as has at least $n/3$ outneighbors, does D always contain a directed triangle (a circuit of size 3) ?

The Union-Closed Sets Conjecture

This conjecture is from Frankl, who in [81] wrote it had first been formulated in 1979.

Given a union-closed set system, i.e. a family of sets $\mathcal{F} \subseteq 2^X$ such that $\forall S_1, S_2 \in \mathcal{F}$ the union $S_1 \cup S_2$ is also an element of $\mathcal{F}$, does there always exist an element $x \in X$ such that x appears in at least one half of the elements of $\mathcal{F}$?

The Erdős-Hajnal conjecture

This conjecture is the most recent of all those, having been first published in 1989 by Paul Erdős and András Hajnal [71].

Given a graph H and the family of graphs $\mathcal{G}$ such that no $G \in \mathcal{G}$ contains an induced subgraph isomorphic to H, does there exists a constant $c_H > 0$ such that the elements of $\mathcal{G}$ contain either a maximum independent set of size $\Theta(n^{c_H})$ or a maximum clique of size $\Theta(n^{c_H})$?

Equivalently, given a family of graphs $\mathcal{G}$ such that $\forall c > 0$ both the maximum clique and maximum independent set are negligible compared to n^c, does any graph H appear as an induced subgraph of some element of $\mathcal{G}$?

Chapter 0

Coloring-oriented introduction to Graph Theory

In this thesis we will talk about graphs, and graphs are according to the level of abstraction one likes either representations of binary relations, or drawings of dots linked by lines.

They can be used as an abstraction of computer networks, or to describe data structures, and from the very different fields in which they have been reinvented they were enriched with a plethora of optimization and existence problems which gives to their simple structure the depth that could convince somebody to study the object itself. This thesis is the result of the fact that under the layers of applications, the different interpretations of the object, the sometimes heavy formalism and the existing knowledge in graph theory that often blocks the view, there remains a *pretty combinatorial structure* definitely worth studying.

On the formal side, we will stick to the usual definition of graphs, directed graphs, vertices and edges as presented in most textbooks (Berge [25], Diestel [62], Bondy and Murty [34], ...), and attempt to define on the way any notion that may not be standard. The following pages undoubtedly suppose a basic understanding of the elementary results of Graph Theory or of Linear Programming, and any time spent implementing the classical algorithms or actually computing optimal solution to hard graph optimization problems would probably enlighten many of the remarks they contain.

From the necessary theoretical background, no result is to be used as systematically as Hall's Theorem.

Theorem (Hall)**.** *Let G be a bipartite graph with vertex classes A and B. There exists a matching (set of disjoint edges) in G spanning A if and only if*

$$\forall S \subseteq A, |N(S)| \geq |S|$$

As a byproduct, a regular bipartite graphs always admits a perfect matching (i.e. a matching spanning all of its vertices).

This theorem can be seen to appear everywhere in combinatorics, which should come as no surprise given that it actually characterizes the existence of an injective function $f : A \mapsto B$ (bijective when $|A| = |B|$) satisfying a set of constraints of the type $f(a_1) \in N(a_1), f(a_2) \in N(a_2), \ldots$.

Such a function exists unless the set of constraints imply that there should exist an injective function from a set $S_A \subseteq A$ to a set $S_B \subseteq B$ of strictly smaller size – which is naturally impossible. Such a wide expressivity is often the key to many beautiful proofs and algorithms.

Of equal importance, dispite being formally stronger (it implies Hall's theorem) is Menger's theorem on the duality between maximum flow and minimum cut.

Theorem (Menger)**.** *Given a graph (or a digraph D) containing two vertices s and t, the maximum value of a flow from s to t is equal to the minimum size of a set of edges (or arcs) intersecting all the paths from s to t.*

These results are especially powerful in the fact that they offer at the same time a feasible solution and a certificate of optimality. By Hall's theorem one can exhibit a matching in a given bipartite graph G and the *proof* that no larger matching exists given by a *blocking set*. Through Menger's theorem one can obtain both a flow of value k and a set of k edges disconnecting s and t, as well as the proof that no flow of larger value exists by giving a list of k edges disconnecting s from t.

As dangerous as it can be to hold a conjecture for true while thinking of unsolved problems, this property is acknowledged to be characteristic of polynomial computational problems (until a proof of that conjecture can finally be found). Indeed, problems for which solutions can be encoded and checked polynomially (the problems of the complexity class NP) and those whose certificates of infeasibility have the same property (the problems of the complexity class $coNP$) are conjectured to intersect precisely on computationally easy problems.

Conjecture. $P = NP \cap coNP$

Hall's and Menger's theorems assert the existence of certificates of infeasibility for the Maximum Bipartite Matching problem and the Maximum Flow problem, the two of them being solvable in polynomial time as expected by the previous conjecture.

Much of what follows can be traced back to one of these two theorems, or to the conjecture $P = NP \cap coNP$.

0.1 Several graphs we will need

Much better than defining the object is a list of examples. Among the $2^{\binom{n}{2}}$ different labeled graphs and the roughly equal number of (connected) unlabeled graphs that can be defined on n vertices, graph theoreticians have grouped many into classes for their common properties.

More is actually true : the different graph classes defined – ISGCI[39] lists more than 1200 – are of interest both for their properties and for the questions they reveal. On planar graphs (see p.31), one can define faces and duals, and become interested in properties of a graph that could be traced to its dual (for instance Tait colorings – see Theorem 26 in [104]), and eventually try to generalize them on graphs of higher genus or through matroids. Chordal graphs, though very similar to trees, can be studied with dynamic programming and tree-width in mind. Depending on how lose one likes to be with definitions, one could also claim that random graphs are an independent class of their own : they are a family of graphs – or rather a way to think about them – with its own problems, questions, existence results. From time to time, studies can go so far toward one class of graphs that the results obtained on them create a new independent theory (see for example Bollobas [31] for random graphs, or Mohar and Thomassen [119] for graphs on surfaces).

As an illustration, we now define two kinds of graphs that will be of use later in this thesis.

Random Graphs

The most common instances of random graphs are the $\mathcal{G}_p^n$, or Erdős-Rényi graphs. With n an integer, and $0 \leq p \leq 1$, one can build a random graph of type $\mathcal{G}_p^n$ by adding independently between each of the $\binom{n}{2}$ available pairs of vertices an edge with a probability p. As any graph H can be obtained through this process with probability $p^{|E(H)|}(1-p)^{\binom{n}{2}-|E(H)|}$ (which equals $1/2^{\binom{n}{2}}$ when $p = \frac{1}{2}$) random graph theory is essentially concerned about properties holding "almost surely" when p is a function depending on n.

These graphs can boast of many properties. Among others, let us say that for a fixed $p \in]0,1[$ the minimum cut of a graph $\mathcal{G}_p^n$ is almost surely the neighborhood of a vertex, or that any graph H is an induced subgraph of $\mathcal{G}_p^n$ with probability 1 when $n \to \infty$. Besides, their *clique number* (the maximum size of a complete subgraph) $\omega(\mathcal{G}_p^n)$ and independent set $\alpha(\mathcal{G}_p^n)$ verify $\omega(\mathcal{G}_p^n), \alpha(\mathcal{G}_p^n) = \Theta(log(n))$ for fixed p, as "implied" by the Erdős-Hajnal conjecture.

Through random graphs have been obtained surprising results of existence (see 0.2.1) – sometimes with embarrassing ease – when deterministic methods had failed beforehand.

Kneser's Graphs

Kneser's graphs can be seen as a generalization of Petersen's graph. They are also highly symmetric (arc-transitive) graphs with various regularity properties, and are defined by two integer parameters n and k.

Kneser's graph K_k^n is the graph built on the $\binom{n}{k}$ subsets of size k of $[n]$, two of them being adjacent when they are disjoint. For this reason, one often sees the additional constraint $2k \leq n$, without which K_k^n has no edges.

Petersen's graph is isomorphic to K_2^5.

0.2 Graph Coloring

This section will revolve around the notion of proper coloring, and use it as an excuse to present several proof methods.

As one of the most basic graph-theoretic notion, the first strength of graph coloring is generality. Many problems often lead to, or can be rephrased, as coloring problems, and any knowledge related to this area is bound to have a wide range of (at least theoretical) applications.

Its most usual definition – which is probably the least natural – is the following : a *proper coloring* of a graph G is a function $c : V(G) \mapsto \{1, 2, \dots\}$ such that $c(u) \neq c(v)$ whenever u and v are adjacent.

Of course, to any finite graph one can associate such a function by assigning to each vertex an exclusive number. When dealing with coloring problems, one is usually interested in knowing the

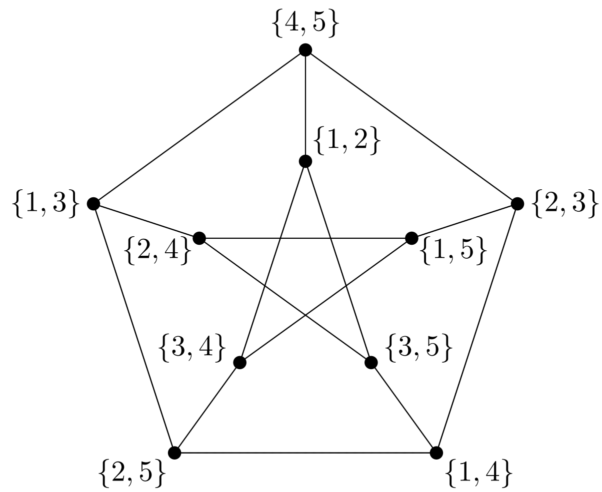


Figure 1: Petersen's graph as one of Kneser's graphs

least value k such that $c(v) \leq k$ for any vertex. This number is called the *chromatic number* of G, written $\chi(G)$.

This definition of graph coloring – or of the parameter χ – can be rephrased in several different ways :

- $\chi(G)$ is the smallest cardinality of a partition of $V(G)$ into independent sets (i.e. sets of pairwise non-adjacent vertices).
- $\chi(G)$ is the smallest integer k such that G can be written as (a subgraph of) a k-partite graph.
- $\chi(G)$ is the smallest integer k such that there exists a homomorphism from G to K_k, the complete graph on k elements.
- Let (A, B) denote, for $A \sqcup B = V(G)$, the set of edges between A and B. Then the smallest number k such that the edges of G can be written as $(A_1, B_1) \cup \cdots \cup (A_k, B_k)$ is equal to $\lceil \log_2(\chi(G)) \rceil$.

To some extent, the following results could be admitted as alternative definition of χ : even though they appeared later, they both make this parameter appear as the solution of an elementary graph problem.

- The Hasse-Gallai-Roy-Vitaver theorem ([97, 83, 138, 151]) : $\chi(G)$ is the smallest integer k such that there exist an orientation of G whose longest directed path has k vertices.
- The Erdös-Stone theorem[72] : for any non-bipartite graph H, the maximum density of a H-free[1] graph is
$$\left(\frac{\chi(H) - 2}{\chi(H) - 1} + o(1)\right) \binom{n}{2}$$

[1] A graph is H-free if it does not contain H as a subgraph

Graph coloring is a very elastic notion in many ways. If the very definition has several interpretations, there also exists many paths, many modelings, which lead from a graph or combinatorial problem to graph coloring. It is a good direction to head for whenever one is looking for a "partition of a set into well-behaved sets" – some modeling may then be sufficient to find an equivalent coloring problem.

Here is an illustration of coloring problems converging toward Baranyai's wonderful theorem (see, for example, Jukna's book on extremal combinatorics [107]):

Theorem. *Let X be a set of cardinality n. If k divides n, then the set $\binom{X}{k}$ of all k-subsets of X can be partitioned into partitions of X.*

Theorem. *Let X be a set of cardinality n. If there exists integers α_1, α_2, d such that*

$$n = \alpha_1 k_1 + \alpha_2 k_2 \text{ and } d\alpha_1 k_1 = \binom{n}{k_1} \text{ and } d\alpha_2 k_2 = \binom{n}{k_2}$$

then the set $\binom{X}{k_1} \cup \binom{X}{k_2}$ can be partitioned into d partitions of X, each of which contains α_1 k_2-subsets and α_2 k_2-subsets.

Baranyai's theorem is actually a powerful generalization of these results, going further into the same direction.

Theorem. *Let X be a set of cardinality n, integers $k_1, \ldots, k_r$ and $(a_{i,j})_{\substack{1\leq i\leq r\\ 1\leq j\leq s}}$ such that*

$$\forall i, \sum_{1\leq j\leq s} a_{i,j} = \binom{n}{k_i} \text{ and } \forall j, \sum_{1\leq i\leq r} a_{i,j} = n$$

Then the set $\binom{X}{k_1} \cup \cdots \cup \binom{X}{k_r}$ can be partitioned into partitions of X such that the jth partition contains $a_{i,j}$ sets of cardinality k_i.

This theorem can be rephrased as a coloring problem by very simple means : defining a graph G whose vertices are the elements of $\binom{X}{k_1} \cup \cdots \cup \binom{X}{k_r}$, two of which being adjacent whenever their intersection is nonempty. In such a graph, an independent set corresponds to a family of pairwise non-intersecting sets – note that it is not necessarily a partition of X, as some elements of X could not be covered. Hopefully, missing vertices will probably mean using more classes in the long run. Partitioning $\binom{X}{k_1} \cup \cdots \cup \binom{X}{k_r}$ into partitions of X is possible if and only if $\chi(G) = \frac{1}{n}\left(k_1\binom{n}{k_1} + \cdots + k_r\binom{n}{k_r}\right)$ (which ensures that each independent set of a the coloring covers precisely n vertices).

This very pleasant expressivity naturally motivates research in graph coloring, at first following the other problems being reduced to coloring, and then ... for itself !

0.2.1 $\chi(G)$ seen from other parameters

The computation of $\chi(G)$ is one of the earliest NP-Hard problems. Besides being hard to compute, the chromatic number of a graph is also hard to approximate by less than a $O(n^{1-\epsilon})$ factor unless $P = NP$ [159]. Together with the conjecture $P = NP \cap coNP$ (see p.12) we are led to think that there exists no polynomial certificate – no graph decomposition, no nice characterization[2] – giving a polynomially close idea of the value of $\chi(G)$; that there is, in fact, no polynomial algorithm able to produce a k coloring of a graph G or to certify that $\chi(G) > k^{1-\epsilon}$, i.e. "almost" a proof of infeasibility. By itself, hardness to approximate χ also means that any graph parameter that can be polynomially computed would at times be polynomially far from χ.

Without supposing the truth of any complexity-related conjecture, let us look at some graph parameters and see how they relate to χ.

Maximum degree $\Delta(G)$

The maximum degree $\Delta(G)$ of a graph is actually a natural bound for the chromatic number, and one can obtain a proper coloring of a graph G using at most $\Delta(G)+1$ colors by using the greedy coloring algorithm.

Greedy coloring algorithm

Iterate over the vertices of G, and assign to a vertex v the smallest positive integer which is not already assigned to one of its neighbors, in order to keep the current coloring proper.

At most $\Delta(G) + 1$ colors are necessary for such a coloring.

By Brooks' theorem [40], $\Delta(G) + 1$ colors are required in only two very specific situations, and it is for this reason often safe to assume that $\chi(G) \leq \Delta(G)$, and to deal independently with the remaining cases if necessary.

Theorem (Brooks (1941))**.** *For any connected graph G holds* $\chi(G) \leq \Delta(G)$*, unless G is a complete graph or an odd cycle, in which case* $\chi(G) = \Delta(G) + 1$.

The greedy coloring algorithm yields very different results according to the order in which the vertices are colored. On lucky days, it can return a coloring with $\chi(G)$ colors[3], but at other times return a coloring using $\Delta(G)+1$ colors instead, even when G is a forest (hence bipartite). Determining the worst behavior of the greedy algorithm is actually a hard problem by itself, and evaluating $\Gamma(G)$ – that is the worst possible performance of the greedy algorithm on a given graph – computationally hard. While Brooks' theorem and its (constructive) proof suffices to identify the graphs such that $\chi(G) = \Delta(G) + 1$ and obtain a coloring with less than $\Delta(G)$ colors otherwise, it is already NP-Hard [98] to determine whether $\Gamma(G) = \Delta(G) + 1$.

[2] This conjecture actually prevents the existence of any "polynomial" decomposition, i.e. a decomposition of non-k-colorable graphs that could be encoded in polynomial space and attest that a given graph is not k-colorable. This being said, there actually exists a graph decomposition characteristic of non-k-colorable graphs given by Hajos' theorem (see Jensen and Toft's book [104]), which builds all non-k-colorable graphs from the complete graph on k elements and three simple operations. It is still open, however, to determine whether the length of a decomposition of a non-k-colorable graph G could be exponential compared to the size of G (see [130]).

[3] To do so, one can first compute a proper coloring of a graph G into $\chi(G)$ color classes $I_1, \dots, I_{\chi(G)}$, then apply the greedy algorithm to G by first greedily coloring the vertices from I_1, then the vertices from I_2, etc ... If it does not give a way to compute an optimal coloring, it proves at least that such a coloring *can* be produced by the greedy algorithm on a specific vertex ordering.

In practice, the greedy algorithm is so quick that it can be profitable to run it several times with different orderings, in order to obtain good upper bounds on χ. The best upper bound can then be fed to an exact algorithm for graph coloring, which can use it to cut its exploration tree.

Finding good orderings is a problem with many heuristics, and randomizing them often a useful idea. The most commonly implemented method is perhaps to find an ordering on the vertices minimizing the number of *already colored* neighbors over all the vertices : the trivial bound $\chi(G) \leq \Delta(G)+1$ is obtained through noticing that a vertex v has at most $d_G(v)$ neighbors whose color has already be picked, but there are graphs for which it is possible to order the vertices in such a way that the number of colored neighbors is always far below $\Delta(G)$.

In particular, it is possible to find in planar graphs (see p.31) an ordering such that each vertex has at most 5 colored neighbors before it is colored itself, ensuring that $\chi(G) \leq 6$. Such an ordering is produced by iteratively removing from the graph a vertex a minimum degree, and applying the greedy coloring algorithm on the reverse of this elimination order. With this algorithm in mind, one can associate to each graph G an integer $\delta^*(G)$ – the *degeneracy* of G – corresponding to the maximum degree of a vertex removed during this procedure. It satisfies the equality

$$\delta^*(G) = \max_{\substack{H \subseteq G \\ \text{a subgraph}}} \min_{v \in H} d_H(v)$$

This yields the following bound on the chromatic number of a graph.

Theorem. *For any graph G holds the inequality* $\chi(G) \leq \delta^*(G) + 1$.

The reason why $\delta^* \leq 5$ for planar graphs is explained p.35.

Maximum clique $\omega(G)$

One of the earliests remarks one can make about a coloring of a graph is that it also defines a coloring of all of its subgraphs. For this reason the chromatic number is an increasing function over the graphs, when partially ordered by subgraph containment. Hence, any clique of size k in a graph is a sufficient proof that the graph's chromatic number is larger than k. Though the clique number $\omega(G)$ is NP-Hard to compute – and hence no conjecture in complexity theory would prevent us from believing so – knowing the value of $\omega(G)$ still leaves us far from knowing χ. The illustration of it is quickly given by random graphs.

Indeed, an easy upper bound can be obtained on the size of a maximum clique in the average graph $\mathcal{G}_{n,1/2}$ by the so-called first moment method [9]. The expected number of k-cliques in $\mathcal{G}_{n,1/2}$ is $\binom{n}{k}2^{-\binom{k}{2}}$. Besides,

$$\binom{n}{k}2^{-\binom{k}{2}} \leq n^k 2^{-\binom{k}{2}} \leq 2^{k\log_2(n)-\binom{k}{2}}$$

Hence, the probability that a graph $\mathcal{G}_{n,1/2}$ has at least one clique of size k (and so any clique larger than k) when $k = 2\lceil\log_2(n)\rceil + 3$ is at most

$$2^{k\log_2(n)-\binom{k}{2}} \leq 2^{k\log_2(n)-\frac{1}{2}k(2\log_2(n)+2)} \leq 2^{-k}$$

As this analysis can be repeated to prove the very same bound on the size of the maximum independent set in $\mathcal{G}_{n,1/2}$, such graphs constitute an infinite family whose maximum clique is of logarithmic size and whose chromatic number is at least $\frac{n}{2\log_2(n)}$ by virtue of the general inequality $\chi(G) \geq |V(G)|/\alpha(G)$, where $\alpha(G)$ denote the size of a maximum independent set of G.

Girth $g(G)$

If it is a fact that small cliques do not necessarily indicate a small chromatic number, one may still wonder whether some other known measures of complexity may be of any influence on the value of χ.

The *girth* $g(G)$ of a graph G – that is the size of a smallest cycle in G – is a perfectly good candidate : around each vertex v of a graph with high girth one can think of G as a tree, and one may be tempted to think that such an information could be strong enough to ensure a small chromatic number.

Erdős proved in 1959 that it is not so [68], and that there exists for any value of g and k a graph whose girth is at least g and chromatic number at least k. The proof, which follows (and can be be read in "The Probabilistic Method" [9]), is again based on a random graph $\mathcal{G}_p^n$, from which several vertices are removed to ensure a girth of at least g.

In a graph $\mathcal{G}_p^n$, the average number of cycles of length at most g is

$$\sum_{3 \leq i \leq g} p^i \frac{n \times \cdots \times (n-i+1)}{2i}$$

If we intend to remove vertices from this graph to ensure a girth of at least g, we should first make sure that it is not necessary to remove *all* the vertices. Removing a small part of them would be preferable, and so we would like to choose p and g so that this expected value is small compared to n. For a fixed g, this could lead us to require that $p = n^{1/g-1-\epsilon}$, as

$$\sum_{3 \leq i \leq g} p^i \frac{n \times \cdots \times (n-i+1)}{2i} \leq \sum_{3 \leq i \leq g} \frac{(pn)^i}{2i} \leq g(pn)^g \leq gn^{-\epsilon}$$

Hence, with this choice of p one would have to remove an average of at most $gn^{-\epsilon}$ vertices from $\mathcal{G}_p^n$ to ensure it is of girth at most g. Besides, the average maximum independent set of such a graph is – as previously – asymptotically small compared to n.

$$\binom{n}{c}(1-p)^{\binom{c}{2}} \leq \left(n(1-p)^{(c-1)/2}\right)^c \leq (ne^{-p(c-1)/2})^c$$

And there is almost surely no independent set of size

$$c = 1 + (3/p)log(n) \leq n^{1+2\epsilon-1/g} = o(n) \text{ assuming } 2\epsilon < 1/g$$

in such a random graph. As a consequence of these two facts, there exists a random graph with these parameters from which removing $gn^{-\epsilon}$ vertices produces a graph G of girth g, in which $\alpha(G) = o(n)$, leading to $\chi(G) \approx n/log(n)$ and to the desired result.

This means in particular that for any graph H containing a cycle, there are H-subgraph-free graphs of arbitrarily large chromatic number. On the other hand, there exists for each tree T an integer c_T such that every T-subgraph-free graph has a chromatic number at most c_T [146, 93].

This being said, one should not – for so few – lose all hope of comparing ω to χ. If it is indeed true that no function of ω is a bound on χ, or that given a tree T there exists T-free (for induced subgraph inclusion) graphs of arbitrarily large chromatic number (for example a large complete graph), there could well be a balance between the two. When T is a tree, Gyàrfàs [92] and Sumner [146] conjectured that in the class of T-free graphs the value of χ can be upper-bounded by a function of ω. This conjecture is mostly open today, though proved in the restricted case where H has diameter two [109].

Fractional Chromatic number $\chi_f(G)$

The fractional chromatic number is a lower bound on the chromatic number which appears when one chooses χ to be the "least number of independent sets covering all the vertices". In this setting, the problem of finding a smallest family of independent sets covering the vertices can be relaxed by associating to each independent set S of the graph a real value w_S, and requiring that each vertex be "covered" in this relaxed sense, namely that

$$\sum_{\substack{v \in S \subseteq V(G) \\ \text{an independent set}}} w_S \geq 1$$

The solutions of this problem when the values w_S are integers instead of real values are precisely vertex colorings.

This alternative formulation is precisely made to fit into the framework of Linear Programming, and results in the following two formulations.

Fractional Chromatic Number

- Minimize

$$\sum_{S \text{ independent set } \subseteq G} w_S$$

- Such that :

$$\forall v \in G, \sum_{\substack{S \text{ independent set } \subseteq G \\ v \in S}} w_S \geq 1$$

Or its dual

Fractional Chromatic Number (Dual)

- Maximize

$$\sum_{v \in G} x_v$$

- Such that :

$$\forall S \text{ independent set } \in G, \sum_{v \in S} x_v \leq 1$$

Computing this value has been proven NP-Hard [89], and is still hard to approximate within a polynomial factor [114]. Numerically, $\chi_f(G)$ can be arbitrarily far from the exact chromatic number of a graph, and is always at least as strong as the bound $\chi(G) \geq n/\alpha(G)$.

This is illustrated by Kneser's graph (see p.13) K_k^n. Kneser himself conjectured in 1955 that $\chi(K_k^n) = n - 2k + 2$, which was in 1978 proved by Lovász using a amazing argument from topology [113, 116]. On the other hand, the inequality $\chi_f(K_k^n) \leq n/k$ is obtained by giving a weight of $1/k$ to each stable set $\{s \in \binom{n}{k} : i \in s\}$ for all $i \in [n]$.

0.3 Choosability

0.3.1 Definition

Formally speaking, the definition of the choosability of a graph is very close to the definition of its chromatic number. Up to now, the vertices had to be colored properly with integers from a set of cardinality k, and choosability changes to vertex coloring that the vertices may be assigned colors from different sets.

Given a function $L : V(G) \mapsto 2^{\mathbb{N}}$, a graph G is said to be *L-list-colorable* if there exists a proper assignment of colors to the vertices (a function $c : V(G) \mapsto \mathbb{N}$) such that each vertex receives a color allowed by its list ($\forall v, c(v) \in L(V)$). To associate an integer *measure* to this version of vertex coloring which could be comparable to its "number of available colors", the size of lists is then bounded by a common value : a graph is said to be *k-choosable* if it can be properly L-list-colored for any choice of a function L satisfying $L(v) \geq k, \forall v$ (which is actually equivalent of being list-colorable for any choice of lists of size exactly k). The *choosability* of a graph G, written $ch(G)$, is equal to the smallest integer k such that G is k-choosable.

From the definition, it appears that k-choosable graphs also are k-colorable graphs (as they can be colored with a constant list function equal to $\{1, \ldots, k\}$). Besides, the proofs of results based on local considerations can sometimes be lifted from vertex coloring to vertex choosability : in particular, the inequality $ch(G) \leq \delta^*(G) + 1$ holds.

One of the classical proofs of Brooks' theorem, however, for being based upon recoloring arguments, does not immediately transfer to graph choosability. This being said, the result remains valid in the context of choosability by virtue of a characterization of *degree-choosable* graphs.

Theorem (Borodin [35], Erdős, Rubin, Taylor [70])**.** *A graph G is said to be degree-choosable if it is L-list-colorable for any function L associating to each vertex $v \in V(G)$ a list of $d_G(v)$ admissible colors.*

Equivalently, a graph G is degree-choosable if and only if all of its 2-connected components are complete graphs or odd cycles (i.e. G is a Gallai tree, see Fig.2).

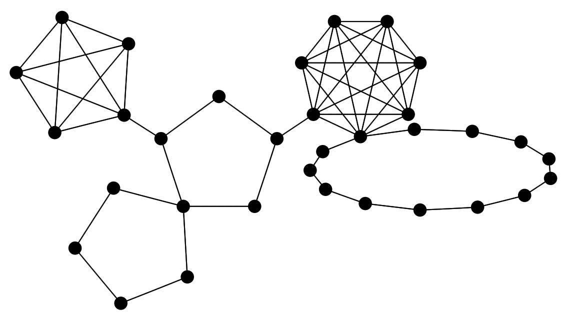


Figure 2: A Gallai Tree

As a corollary of this result, the only graphs verifying $ch(G) = \Delta(G) + 1$ are those that verified the same relationship for χ, namely odd cycles and complete graphs [148]. This is actually one of the rare occurrences for which the two parameters meet, and the difference between the two parameters appear as soon as $k = 2$: very few bipartite (2-colorable) graphs are 2-choosable.

Theorem (Erdős, Rubin, Taylor [70])**.** *A (connected) graph is 2-choosable if and only if it can be reduced to the empty graph by iteratively removing vertices of degree one – i.e. the graph is a tree – or if what remains is an even cycle, or the union of a C_4 and another even cycle intersecting on three consecutive vertices (see Fig.3).*

Even more striking is the fact that bipartite graphs – and in particular complete bipartite graphs – may have an arbitrarily large choice number, as shown by the following classical construction.

Bipartite graphs

Let $k > 0$ be an integer, and G be the complete bipartite graph defined on two disjoint copies of the family $\binom{2k}{k}$ of all k-subsets of a set of cardinality $2k$, each vertex being assigned as a list the k elements to which it corresponds. As the graph is complete bipartite, the set of integers used to color the vertices of one copy is disjoint from the set used to color the other vertices, ensuring that one of

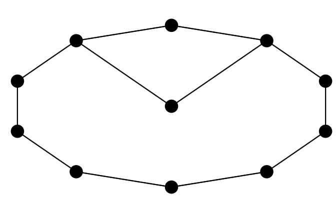
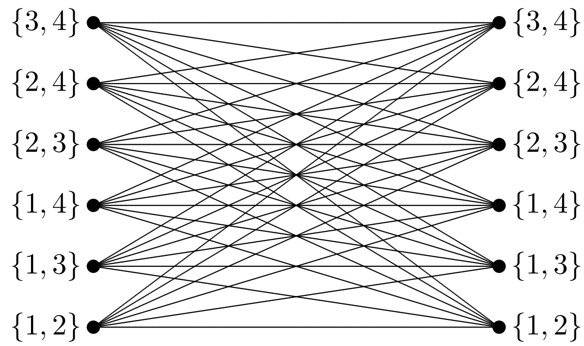


Figure 3: A C_4 and an even cycle, intersecting on three consecutive vertices.



them is of size at most k. On the other hand, it is not possible to color vertices corresponding to all the k-subsets with only k colors, as there is a set of size k in the set of size $2k$ which does not contain any of them.

Certificates

Another important difference between chromatic number and choice number is that the first – even though NP-Hard to compute – belongs to NP : the easiest way to certify that a graph is k-colorable is to produce a proper k-coloring, and checking that a coloring is proper is easily done. On the other hand, certifying that a graph is k-choosable is giving a proof that to any list assignment can be associated a satisfiable proper coloring[4] which seems computationally much harder [91]. For this reason, any general framework able to prove an upper bound on the choice number of a graph is particularly precious[5].

0.3.2 Nullstellensatz - Alon Tarsi - Bipartite graphs

Alon and Tarsi [11] introduced in 1992 a surprising and elegant method to prove an upper bound on the choice number of a graph. It is algebraical, and based on Hilbert's Nullstellensatz which asserts that if any nonzero real polynomial of degree d has at most d roots, the same can be said about multivariate polynomials defined over a field.

Theorem (A first version of the Combinatorial Nullstellensatz [11]). *Let $P = P(x_1, \dots, x_n)$ be a polynomial of $\mathbb{Z}[X_1, \dots, X_n]$ whose degree in x_i is at most d_i. Then, for any choice of $S_i \subseteq \mathbb{Z}$ such that $|S_i| > d_i$, P can not be equal to zero on the elements of $S_1 \times \cdots \times S_n$ unless $P \equiv 0$.*

In "The combinatorial Nullstellensatz", Alon [5] presents several combinatorial problems which can be formulated in the formalism given by this theorem (as graph-theoretic optimization or existence problems can be formulated using the formalism of linear programming or submodular functions). This approach leads one to encode the data of a problem into a nonzero polynomial of smallest possible degree whose zeroes are infeasible or *bad* solutions, on which the Combinatorial Nullstellensatz

[4]This problem is finite : given a graph on n vertices and an integer k, the number of possible assignations of lists of size k is infinite as any imaginable color could be used in the lists, though what really matters is how the lists intersect each other. In particular, a list assignment using a total of nk colors would be giving disjoints sets to each vertex (of course, many other different list assignments produce identical intersection patterns between the lists). However, this information would just let us bound the number of assignment by $\binom{nk-1}{k}^k$ which makes absolutely no sense for any practical application.

[5]For instance, the Alon-Tarsi method presented p.22, or Galvin's proof on the choice index of bipartite graphs p.30.

can finally be used to prove the existence of a *good* solution [6].

This being said, this method has the unpleasant idea to only provide non-constructive proofs of existence. The second regret is that such a tool can not be hoped to always be sufficient as it is an implication and not an equivalence – hence one can not hope to obtain short proofs of infeasibility as in the case of Linear Programming and its duality theorem.

One of its remarkable consequences deals with coloring, as presented in [11].

Choosability through the Nullstellensatz

Given a graph G, one can quickly encode a feasible assignment of colors as coordinates for which a specific polynomial is non-zero. It would require to associate a variable $x_v \in \mathbb{N}$ to each vertex and study the solution of any non-zero value in

$$\prod_{v \in G} \prod_{u \in N_G(v)} (x_u - x_v)$$

What can the Combinatorial Nullstellensatz tell us at this step ? One product being included in the other, each variable x_w appears $d_G(w)$ times when $v = w$, and once for every neighbor or w. Hence, we know that if vertex v is assigned a list of $2d(v) + 1$ colors, then there exists a satisfiable list-coloring which implies $ch(G) \leq 2\Delta(G) + 1$. This should not come as a surprise, as a greedy coloring gives us $ch(G) \leq \Delta(G) + 1$ and even $ch(G) \leq mad(G) + 1$. This polynomial, though, contains for each edge uv both the factors $(x_u - x_v)$ and $(x_v - x_u)$, which tells us to focus instead on its "square root" [7].

$$\prod_{uv \in G} (x_u - x_v)$$

This new polynomial tells us that if each vertex v is associated to a list of size at least $d(v) + 1$ a feasible assignment necessarily exists, implying the inequality $ch(G) \leq \Delta(G) + 1$. This is nice, but we should expect much better, such a different approach of coloring is bound to yield new and surprising results !

To improve it, this stronger version of the previous theorem is required.

Theorem (A second version of the Combinatorial Nullstellensatz [11])**.** *Let $P = P(x_1, \dots, x_n)$ be a polynomial of $\mathbb{Z}[X_1, \dots, X_n]$ whose degree is $d = \sum_i d_i$. If the coefficient in P of $x_1^{d_1} \dots x_n^{d_n}$ is nonzero, then for any choice of $S_i \subseteq \mathbb{Z}$ such that $|S_i| > d_i$, P can not be equal to zero on the elements of $S_1 \times \cdots \times S_n$ unless $P \equiv 0$.*

This second formulation leads us to focus on the shape of the monomials in the polynomial's expansion. The whole polynomial's degree is obvious : each factor is of degree 1 and their number is $|E|$, hence the polynomial itself is of degree $|E|$, and so its monomials of maximum degree are those obtained by picking one of the two variables in each of its factors.

This is precisely the definition of an *orientation* of a graph.

[6] Anybody enjoying this paper will probably take pleasure in reading "Non-constructive proofs in combinatorics" [4] from the same author.

[7] Note that this polynomial is not properly defined, as taking vu instead of vu in the enumeration of the edges would change its sign, though being only interested in its zeroes, any of the two polynomials this could produce is as good as the other for our purposes

If any orientation of the graph results in a monomial of maximum degree, the coefficient of such a monomial can be either 1 or -1, depending on whether the vertex picked in the product $(x_u - x_v)$ is u or v. If we call *positive* an orientation of the graph such that the resulting monomial has coefficient 1, and *negative* when it is -1, the coefficient of $\prod_{v \in G} x_v^{d_v}$ is equal to the difference between the number of *positive* and *negative* orientations such that the outdegree of v is equal to d_v. As we are merely interested in knowing whether this polynomial is nonzero, what concerns us is the answer to :

"are the number of positive and negative orientations of G such that the outdegree of v is d_v equal ?"

Actually, the difference between two given orientations D, D' of a graph such that the outdegree of v is d_v is succinctly described. Let $R \subseteq E(G)$ (for "reversed") be the set of edges whose orientation is different in D and D', and let v be any vertex or G. D' can be obtained from D by reversing all the edges from R (and conversely), and as $d^+_D(v) = d^+_{D'}(v)$ we deduce that the number of arcs *leaving* v one must reverse to obtain D' is equal to the number of arcs *entering* v which are also contained in R.

Hence, the edges of D contained in R constitute an eulerian subgraph of D (the same being true for D'). One can then go from any orientation of G to any other with the same outdegree sequence by reversing the edges of an eulerian subgraph, and any eulerian subgraph of D will result in a different orientation of G. Besides, one can transform a *positive* orientation into a *negative* one (or the opposite) only by reversing an odd number of edges (reversing an even number of edges preserves the *sign* of the orientation). Wondering whether the number of *positive* and *negative* orientations of G with some fixed degree sequence is equal is thus equivalent to answering whether an arbitrary orientation D of G with the same degree sequence has the same number of odd and even eulerian subgraphs.

Bipartite Graphs

Bipartite graphs already have an unbounded choice number as previously explained. At the same time, they are a class of graphs for which the comparison between the number of eulerian cycles is not too troublesome, as having no odd cycle is a sure proof that no odd eulerian subgraph exists. Hence, any orientation D of a bipartite graph G is a certificate that for any list assignment such that v is given a list of $d^+_D(v)$ colors there exists a corresponding list-coloring. By itself, this does not give us any bound on the choice number of a bipartite graph, but the computation of an orientation of a graph whose maximum outdegree is minimal can be done easily by the construction presented on p.33, which is best possible. Hence, the following result :

Theorem (Alon Tarsi (1992) [11])**.** *Let G be a bipartite graph. Then $ch(G) \leq \lceil mad(G)/2 \rceil + 1$*

A game version of choosability

As illustrated by the bipartite graphs, choosability gives a quite different feeling from proper coloring. One of the reasons why it is much harder to work with is probably that proving an upper bound on ch requires to test an intractable collection of list assignments, and the insufferable lack of Kempe-chain-based arguments (see p.28 for a definition of Kempe chains). For this reason, many results dealing with choosability, including those presented in this thesis, content themselves with very local considerations – and amenable classes of graphs, like planar graphs or embeddable ones.

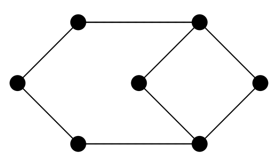
Zhu introduced [158] in 2009 a variant on graph choosability – and playing with the very definition sounds like the sane thing to do in these situations – which, even though heavily more constrained in its rules has not been proved to be significantly harder than graph choosability yet.

In his game two players concur, the two of them aware of the whole graph structure. In turn, the first player will give a subset of the vertices from which the second is to extract a stable set of his choice (he is actually given at time t "the set of vertices containing color t", and has to answer the ones which are to be colored with t). The only added constraint is for the first player to give each vertex at least k times in the sets he produces ("each vertex has a list of k available colors").

If, for a fixed graph and integer k, the second player can always achieve to color all the vertices before all the sets are given, the graph is indeed k-choosable. On the other hand, he may be unable to complete his coloring on a k-choosable graph as the first player decides of the list-assignment while the graph is being colored : the next lists can then depend on the current partial coloring. Hence, the smallest k such that the second player has a winning strategy, noted $ch^{OL}(G)$, is greater than $ch(G)$.

Despite being formally harder than the usual choosability, the difference between the two is far from clear. In particular, the inequality $ch^{OL}(G) \leq \lceil mad(G) \rceil + 1$ holds and dissipates any hope to distinguish ch and ch^{OL} with such arguments. Besides, Zhu also proved that a choosability proof based on the Alon-Tarsi method also transfers to on-line choosability. As the means of proving upper bounds on ch are not that numerous, watching such tools transfer indicates that the difference between the two parameters is not to be witnessed with the state-of-the-art choosability machinery.

In the same paper, Zhu obtains a characterization of graphs with $ch^{OL}(G) = 2$ similar to the one Erdős, Rubin and Taylor obtained from 2-choosable graphs (see p.21), which reveals that some graphs verify $ch \neq ch^{OL}$.



A 2-choosable graph with $ch^{OL} = 3$

Theorem (Zhu (2010)[158]). *A connected graph G satisfies $ch^{OL}(G) \leq 2$ if and only if its 2-core is K_1, C_{2n} or $K_{2,3}$. In particular,* $2 = ch(\theta_{2,2,2}) \neq ch^{OL}(\theta_{2,2,2}) = 3$.

This result leaves open the very interesting question [158] of the possible gap between $ch(G)$ and $ch^{OL}(G)$: is it arbitrary large ? If it is large is $ch^{OL}(G)$ bounded by $c \times ch(G)$ for some c ? By a polynomial function of $ch(G)$?

Being able to bound one using the other would give a new ways to think about the ch parameter, but while these questions stand they point toward ill-understood properties of choosability.

0.4 Edge coloring

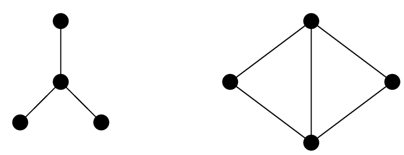
Edge coloring is by definition a coloring on $L(G)$ instead of G itself : one now wants to attribute a color to the edges of a graph in such a way that incident edges have different colors, and $\chi'(G)$ – the *chromatic index* of G – denotes the minimum number of colors required to properly color $L(G)$, i.e. $\chi'(G) = \chi(L(G))$. Edge coloring may not have been deemed worthy of an independent definition if any graph could be represented as the line graph of another one. This is indeed wrong, and line graphs are characterized by a list of forbidden induced subgraphs :

Theorem (Beineke (1970)). *[24] A graph G can be obtained as the line graph of some graph G' if and only if it has no induced subgraph isomorphic to a list of nine small graphs (at most 6 vertices).*

Let G' be a graph whose line graph is G. The vertices of G are the edges of G', and each vertex v of G' produces in G a complete graph, as all the edges incident to v are pairwise adjacent in G. It is then important, when trying to build G from G' (or when trying to detect whether a graph is actually a line graph) to pay attention to the cliques as representant of vertices. Of course, all the maximal cliques of G do not correspond to a vertex of G' – the edges of a triangle also form a triangle in the line graph, which is the only alternative reason for the creation of a clique in a line graph (this becomes for a natural generalization – line graphs of multigraphs).

This is a hint that an easier characterization of line graphs of triangle-free graphs should exist.

Theorem ([41, 99]). *A graph is the line graph of some triangle-free graph G' if and only if it contains no induced subgraph isomorphic to the claw or to the diamond.*



Claw Diamond

Note that the claw already belonged to the list of nine graphs mentioned previously. Naturally, the diamond is an induced subgraph of the eight other graphs.

Vizing's theorem

The obstructions from before (cliques) are only present in line graphs as "the set of all edges incident to a specific vertex". Hence the inequality $\chi'(G) \geq \Delta(G)$. Edge coloring is also equivalent to finding a partition of the edges into matchings. Like vertex coloring, it has a wealth a variants, and the reason why edge-coloring (coloring of line graphs) is a well-studied problem is probably because of Vizing's theorem.

Theorem (Vizing). *[152]* $\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$ *(a proof of this result is given later)*

Besides giving a fair idea of the chromatic index of general graphs (as well as an algorithm, as its proof is constructive), the natural equivalent of the Fractional Chromatic Number (see p.19) – called Fractional Chromatic Index – is a polynomial optimization problem and can in practice help detect graph with $\chi' = \Delta + 1$. The fact this LP is polynomial is a direct implication of the Separation Theorem and Edmond's algorithm for general matching [67]. Moreover, in virtue of the conjecture asserting that for any polynomial decision problem there exists certificates for both True and False answer ($P = NP \cap co-NP$, see p.12) – or more practically because of the LP duality theorem, there exists a clear characterization of graphs with given fractional chromatic index in terms of obstructions by *overfull subgraphs*.

Fractional Chromatic Index

- Minimize
$$\sum_{M \text{ matching } \subseteq G} x_m$$
- Such that :
$$\forall e \in E(G), \sum_{\substack{M \text{ matching } \subseteq G \\ e \in M}} x_m \geq 1$$
- x_m is a real positive variable

Fractional Chromatic Index (dual)

- Maximize
$$\sum_{e \in E(G)} x_e$$
- Such that :
$$\forall M \text{ matching } \subseteq G : \sum_{e \in M} x_e \leq 1$$
- x_e is a real positive variable

Obstructions - Overfull subgraphs

Given the duality theorem on Linear Program, there is some hope that this algorithm gives more than a way to compute the chromatic index but also an idea of what the obstructions are – like Menger's cutset with maximum flow, or Tutte's set for maximum matchings. Indeed, two natural bounds happen to be sufficient. The first is the inequality $\chi'_f(G) \geq \Delta(G)$, and the second an equivalent for edge coloring of the inequality $\chi(G) \geq \frac{|V(G)|}{\alpha(G)}$ giving $\chi'_f(G) \geq \frac{|E(G)|}{m(G)}$ (where $m(G)$ denotes the cardinality of a maximum matching in G).

Theorem (See for example Scheinerman's "Fractional Graph Theory"[140])**.**

$$\chi'_f(G) = max\left(\Delta(G), \max_{\substack{H\subseteq G\\ H\ odd}} \frac{2|E(G)|}{|V(H)-1|}\right) \leq \Delta(G)+1$$

Of course, assuming the eternal $P \neq NP$ conjecture, one can not hope in general that $\lceil \chi'_f(G) \rceil = \chi'(G)$. Hence, there are examples of graphs with $\chi'_f(G) = \Delta(G)$ and $\chi'(G) = \Delta(G)+1$. It is not known – for instance – whether the equality $\lceil \chi'_f(G) \rceil = \chi'(G)$ holds for chordal graphs as it holds for complete graphs. Actually, we are still to find out whether edge coloring is computationally easy on well-known classes of graphs like interval graphs, chordal graphs, and cographs [115].

Kempe chains

We now present a proof of Vizing's theorem, as it is central in edge coloring and can help understand why this powerful result does not easily extend to acyclic edge coloring or edge choosability. It is based upon the use of Kempe chains, or alternating chains, which give a way to modify a given edge coloring of a graph while keeping it proper.

One can obtain a Kempe chain in a given edge-colored graph by picking two adjacent edges e, e'. Being adjacent, their colors are different, and if we write c and c' their two respective colors we can walk through some part of the graph by starting from e or e' and moving from edge to edge by iteratively jumping from an edge of color c to an edge of color c'.

The Kempe chain containing e and e' is actually the connected component containing e and e' in the graph induced by the color classes c and c'. Such a chain can be recolored by transposing the colors c and c' – the coloring obtained remains proper. When it is a path (and not a cycle), the aim of such a transformation is to change the colors available around the vertices at the ends of the path : if one endpoint of a Kempe chain is incident to an edge colored with c (hence it has no incident edge colored with c'), this color will not appear in its neighborhood after the edge-coloring is modified, the corresponding edge being now colored with c' – this can be put to good use to obtain a coloring of the whole graph, which is precisely what Vizing's theorem builds.

Vizing's theorem – Proof (Diestel's Algorithm)

Proof. The proof is constructive, and procedes by coloring G edge by edge, potentially changing the coloring of previously colored edges (but never *uncoloring* any), until all the edges of G are properly colored.

Let the graph G_c (for *current* G) contain as edges those of G which have already been given a color, and let $uv \in E(G)$ be an edge for which we want to find a color in order to extend our partial coloring. As u and v are in G_c of degree at most $\Delta(G)-1$, they both have one (actually, two) color missing in their neighborhood. Let us respectively note a a color which does not appear on the edges incident to v, and name b a color which does not appear around v.

Of course, if a and b are the same color, one can immediately use it to color edge uv, but most probably some other edges of the graph will have to be recolored.

In an attempt to color edge uv with a, it is tempting to consider the maximal ab-path leaving vertex v (i.e. the Kempe chain with colors a and b containing v) : transposing colors a and b on this path has the effect to recolor the edge a around v with color b, which should finally let us color a new edge uv with color a, now free around both u and v. Of course, the maximal ab-path starting from v may very well end at u, in which case reverting the two colors on this maximal ab-path recolors with a an edge around u at the same time that this color becomes available around v.

This is a hopeless situation. We can but stare at v', the last vertex on the ab-path leaving v – which is adjacent to u with an edge colored by b – hoping for some comfort : it also has a free color c, different from both a and b, and it may well be that the ac-chain leaving v' stops very far away in the graph, and not at u once more. In this case, one could hope to recolor this ac-chain by exchanging its two colors. a would now be available around both u and v' which are linked by a edge with color b. This puzzle is then solved by recoloring this edge from b to a, and adding an edge between u and v with color b, as it is now available around both u and v.

Let us remember from this short story that to each vertex $v_1, \dots, v_i$ around u can be associated a free color c_i, as $\Delta(G) + 1$ in total are available. When considering the maximal ac_i-path leaving v_i, we are *sometimes* led to another neighbor of u which can only be $N_u(c_i)$ – the neighbor of u sharing with it an edge colored with c_i, if there is any.

Trying to find a color for edge uv led us to recolor edge $uN_u(a)$, then to recolor edge $uN_u(N_u(a))$, then to $uN_u(N_u(N_u(a)))$ for as the successive Kempe chains guide us through the neighbors of u. This run, however, has to stop somewhere as it can not enter a loop : there can not be two distinct neighbors $v_j, v_{j'}$ of u such that $N_u(c_j) = N_u(c_{j'})$, for this would imply that two distinct alternating chains with the same colors stop at the same vertex. Besides, there is no vertex v_j such that $N_u(c_j) = v$, as the edge between u and v does not belong to the graph just yet. Hence, by starting our run from vertex v we are sure it will stop somewhere.

From the beginning, we actually attempted to give color a to uv, which was only possible if a was available around v. In the latter case, we could still have done so by recoloring a Kempe Chain, but this Kempe chain may not have been interesting to recolor unless the *following* Kempe chain was recolored too. Vizing's method of recoloring the whole graph lies in the fact that his algorithm will at some point reach a Kempe chain that *can* be recolored, meaning that the previous can now be recolored too, and this until the first one, and until uv can be colored too.

With this algorithm, at each step a new edge of G can be recolored, almost greedily, until the whole of G has been assigned a color. □

List coloring conjecture

Even though research is being led on the differences between *class 1* and *class 2* graphs (respectively those whose chromatic index is equal to $\Delta(G)$ or $\Delta(G) + 1$) – even in the restricted case of planar graphs (see p.37) – much about proper edge coloring has been settled by Vizing's theorem. Studying the choosability version of proper edge coloring, on the other hand, led to a conjecture which stands as one of the main open problems in graph coloring. Indeed, while there are clear differences between $ch(G)$ and $\chi(G)$ for general graphs (and even bipartite ones), the conjecture is that the worst assignment of lists to the edges of a graph is the constant one.

Conjecture (List Coloring Conjecture ([32],[94]))**.** *$ch(G) = \chi(G)$ for any line graph G – or equivalently – $\chi'(G) = ch'(G)$ for any graph G.*

This conjecture is especially painful since the proof of Vizing's theorem is very well understood, and relies on simple arguments. Not even the inequality $ch'(G) \leq \Delta(G) + 1$ is known to hold, and one of the most remarkable recent results on the subject is Kahn's one [108] from 2000.

Theorem (Kahn)**.** *For any $\epsilon > 0$ there exists χ'_0 such that any graph G with $\chi'(G) \geq \chi'_0$ satisfies*

$$ch'(G) \leq (1 + \epsilon)\chi'(G)$$

Restricted to specific classes of graphs, the List Coloring Conjecture has been proved to hold on planar graphs whose maximum degree is greater than 12 (see p.37), and on bipartite graphs, a result proved by Galvin [84] that we present below.

Galvin's proof for bipartite graphs – Choosability by kernels

The method with which Galvin [84] proved that for any bipartite graph $ch'(G) = \Delta(G)$ is simple and surprisingly efficient. It is actually the application of a method that Alon and Tarsi [11] attribute to Bondy, Boppana and Siegel, which we now introduce, trying to preserve some intuition.

Given any graph G – which does not have to be a line graph – one can hope to color its vertices with the following greedy algorithm :

Greedily build a maximal independent set, and remove it from the graph for this is a new color class.

With this algorithm, one always obtains a coloring using less than $\Delta(G) + 1$ colors, as at each iteration any vertex which is not included in the current maximal independent set is adjacent to a vertex which is in that set (otherwise, the independent set would not be maximal). If one could ensure, instead, that there is a way to remove, at each turn, an independent set such that any vertex has two neighbors in this set (or is in the set itself), this would directly imply that the algorithm can color a graph using $\Delta(G)/2 + 1$ colors. This is one possible direction, but not the one Bondy, Boppana and Siegel successfully picked.

This algorithm – which for the moment is not much more powerful than the more classical version of greedy coloring – natively works well for list-coloring. Indeed, if any vertex is given a list of colors, one can rewrite it the following way :

Consider the set of vertices whose list contains color 1, and greedily build a maximal independent set with those vertices. Color them with 1, remove them from the graph, and do the same iteratively with the other colors until no vertex remains.

As previously, one can ensure that this algorithm works as soon as the vertices have at least $k = \Delta(G) + 1$ colors available, for if we focus our attention on one vertex v, and the iterations $c_1, \ldots, c_k$ of our algorithm corresponding to the colors contained in v's list, we see that as previously, at each turn either v or one of its neighbors is colored and removed from the graph.

There still remains the fact that such a coloring can not be hoped to do better than $\Delta(G) + 1$.

The Bondy-Boppana-Siegel method is the following : before coloring a graph G, they choose an orientation D of G such that $\Delta^-(D) < k$. Then, not being satisfied with picking a maximal independent set, they attempt to find in D an independent *dominating* set [8] – that is a independent

[8] Also called *kernel* in the literature

set S of vertices independent such that each vertex is in S of has an in-neighbor in S. In the ideal situation where such a set can always be found, the graph G can be colored with a total of k colors, as at each turn either v or (at least) one of its $< k$ in-neighbors is colored.

This technique can be as easily adapted to list-coloring. In this case, its main difficulty lies in finding an orientation D of G with small $\Delta^-(D)$ (for this value corresponds to the number of colors appearing in the final coloring), and such that there exists in any set $S \subseteq V(G)$ of vertices an independent *dominating* set. Such an orientation is the subject of one of Richardson's theorems [135].

Theorem (Richardson (1946))**.** *If a directed graph D does not containing any induced odd circuit, then it is kernel-perfect (i.e. there exists in any induced subdigraph $D' \subseteq_i D$ an independent dominating set).*

In his proof, Galvin [84] shows how to obtain a kernel-perfect orientation D of the line graph of a bipartite graph G with $\Delta^-(D) = \Delta(G)-1$, ensuring that $ch(G) = \Delta(G)$. He builds it by considering the two vertex classes A and B of a bipartite graph, and computing a proper edge coloring (hence, without lists) of G, giving colors $1, \dots, \Delta(G)$ to the edges. The orientation is then built the following way :

- For any pair of incident edges $(u, v), (u', v), v \in B$ with respective colors $c(u, v) < c(u', v)$, there is in D an arc from (u, v) to (u', v)
- If $v' \in A$, the arc from (u, v) to (u', v) is created if $c(u, v) > c(u', v)$ (the constraint is reversed)

As there are in G a total of $\Delta(G)$ colors, in the given orientation D no vertex (corresponding to the edges of G) has indegree more than $\Delta(G) - 1$. We are left with the task to prove that there exists in any induced subdigraph $D' \subseteq_i D$ an independent dominating set, which actually is a direct consequence of the Stable Marriage Theorem[9]. Indeed, in D' is defined for each vertex v an ordering on the edges of G incident to v (and hence an ordering on the *neighbors* of v) : this linear ordering, seen as a set of preferences, defines a Stable Marriage Problem whose aim is to find a matching in G such that any other edge of G is dominated in D'. The existence of a stable marriage completes the proof.

0.5 Planar graphs

A very large part of the work presented in this thesis focuses on *planar* graphs. A planar graph is nothing more than a graph that can be drawn on the plane without two edges crossing, or equivalently a drawing of a graph in which two edges intersect only at their common endvertices. Planar graphs are a frequent occurrence of graphs arising from practical problems, and have been completely characterized [111] in 1930 by Kuratowski in terms of *forbidden subgraphs*.

[9]The Stable Marriage theorem is *the* actual contribution from Graph Theory to human happiness, and no one short of graph theoreticians really seems to care. It incidentally deals with matchings in bipartite graphs. In the usual setting, we have in front of us a set of n women and n men, all very eager to get married. Being so keen on putting an end to celibacy as soon as possible, they all have sorted their possible mates in a linear order, according to their preferences. A Stable Marriage is then defined as a matching such that no two people would prefer to be married together than to their actual mate, hence preventing the chance of any "future troubles". Scientists noticed that non-stable marriages usually make things more complicated, and hopefully the Stable Marriage Theorem asserts that a stable marriage always exists (and is computationally easy to find).

Theorem (Kuratowski)**.** *A graph G is planar if and only if it does not contain as a subgraph a subdivision of K_5 or a subdivision of $K_{3,3}$.*

Hence, testing whether a graph is planar is a computational problem of $coNP$, as a certificate – in this case, a subgraph – can be given as a proof that it is *not* planar. It is obviously a problem of NP, as a planar embedding of a graph is a sufficient proof that it is planar. Deciding whether a graph is planar can be done in linear time (see [38]), which provides yet another illustration of the conjecture $P = NP \cap coNP$.

Because planar graphs are sparse (their average degree is strictly less than 6), because they admit several equivalent characterizations (cycle space, minors, intersection of linear orders [142], ...), because they can be easily drawn and visually worked upon, because there is a notion of left and right, and thanks to the whole body of the theory of planar graphs that has been built during the past decades, many theoretical or practical, existential or optimization problems are found to be easier on planar graphs than on general graphs or other graph classes.

This does not mean either that there are no complex problems on planar graphs : it is already NP-hard to compute the chromatic number of a 4-regular planar graph [59], and plenty of other examples are to be found all around.

In order to understand the origin of this complexity arising both in computational and purely graph-theoretical problems, researchers naturally attempted to define various graph parameters that would stay meaningful on planar graphs in order to capture it.

0.5.1 Maximum Average Degree

The maximum average degree is one such example.

The average degree of a graph – $ad(G) = \frac{2|E(G)|}{|V(G)|}$ – gives some information on its density, but to make any use of it one has to consider the graph as a whole and can not infer any *local* property – for example on the neighborhood of a vertex. Indeed, any graph H can appear as an induced subgraph of a graph with high (or low) density : it is enough to add to H a big complete (or independent) graph, so as to change its density at will.

The maximum average degree of a graph, denoted $mad(G)$, is a slight modification of the average degree. It is by definition greater than the average degree and gives on planar graphs an appreciable *local* information.

Definition (Maximum Average Degree)**.** The maximum average degree of a graph is the density of its densest subgraph. Formally

$$mad(G) = \max_{H \subseteq G} \frac{2|E(H)|}{|VH|}$$

This parameter can be quickly computed by linear programming (see p. 34).

Hence, a graph G has $mad(G) \leq c$ if and only if none of its subgraphs has density $> c$, which already is an appreciable local information. For this reason, the maximum average degree can be used to guide inductive proofs, as it is increasing according to the usual graph containment ordering. Removing a vertex from a graph of known mad can only decrease its value, while the same is immediately seen to be wrong for the average degree.

Besides, one can notice that many inductive proofs – for instance on planar graphs – can be lifted at no cost to equivalent proofs on the larger class of graphs with $mad < 6$ which contains it.

Finally, maximum average degree and degeneracy (see p.17) encode similar information. As, from the definition, any subgraph H of G has average degree at most $mad(G)$, then any subgraph H contains a vertex of degree at most $\lfloor mad(G) \rfloor$. As a result, it is possible to reduce the whole graph G to the empty graph by iteratively removing vertices of degree at most $\lfloor mad(G) \rfloor$, and the inequality $\delta^*(G) \leq mad(G)$ holds. In the opposite direction, as the number of edges in a δ^*-degenerate graph is at most $\delta^* n$ (and the same holds for any subgraph), its average degree is at most $2\delta^*$. Hence, there also holds $\delta^*(G) \leq mad(G) \leq 2\delta^*(G)$.

These two bounds are tight. Indeed, any k-regular graph G verifies $mad(G) = \delta^*(G)$, and for any $k > 0$ the k^{th} power of a path on n vertices P_n^k – i.e. the graph defined on $\{1, \dots, n\}$ in which $i \sim j$ if $|i - j| \leq k$ – verifies asymptotically $\lim\limits_{x \to +\infty} mad(P_n^k)/\delta^*(P_n^k) = 2$

Algorithmically

It is also good to consider as an "alternative definition" of the maximum average degree the following result.

Theorem. *Any graph G has an orientation D such that $\Delta^+(D) = \lceil mad(G)/2 \rceil$.*

This theorem is perfectly tight : if a graph G contains a subgraph $H \subseteq G$, any orientation of G is an orientation of H, and as a result the maximum outdegree of an orientation is greater than the average outdegree in H, namely $\lceil ad(H)/2 \rceil$. Applied over all the subgraphs of G, this leads to $\Delta^+(G') \geq \lceil mad(G)/2 \rceil$.

■ How to obtain an orientation D of a graph G with $\Delta^+(D) \leq \lceil mad(G)/2 \rceil$? ■

By reducing it to a bipartite flow/matching.

Let B be a bipartite graph on sets L and R, L being equal to $E(G)$ and R containing $d = \lceil mad(G)/2 \rceil$ distinct copies of each vertex. Let it also contain an edge between any element of $uv \in L$ and any copy of u or v in R

This graph admits a matching saturating the set L : the neighborhood of any set $S \subseteq L = E(G)$ has a cardinality of at least d times (number of copies for each vertex) $2|S|/mad(G)$ (from the maximum density of a subgraph in H). Hence $|N_B(S)| \geq \lceil mad(G)/2 \rceil 2|S|/mad(G) \geq |S|$. By Hall's theorem it is possible to associate to each edge one of the copies of its endpoints.

One can then orient each edge of G **away** from its corresponding endpoint in the matching. As there are $\lceil mad(G)/2 \rceil$ copies of each vertex, this orientation has the desired maximum outdegree.

■ ■

Despite its formal definition, the maximum average degree of a graph – and the corresponding densest subgraph which is its certificate – can be computed in polynomial time through linear programming. This program is actually a non-integer version of the matching/flow algorithm just presented.

Computing the maximum average degree by Linear Programming

In order to compute the maximum average degree of a graph G, we will define a directed bipartite graph B similar to the previous one. Its left set L is still equal to $E(G)$, but the other side R is now equal to $V(G)$. An element $uv \in E(G)$ is the origin of two arcs towards its endpoints, u and v, and all these have a capacity of 1. Added to this graph, a source s with arcs toward any element of L – those are of capacity 1 – and a sink t with corresponding arcs from any element of R to t, these last arcs having an unknown capacity c[10].

We are interested in the value of a maximum st-flow in this graph, depending on c. By Menger's theorem, it is equal to the weight of a minimum st-cut $\mathcal{C} \subseteq E(B)$. We would now like to prove that a cut $\mathcal{C}$ corresponds to an induced subgraph of G.

If $\mathcal{C}$ contains an edge ev where $e \in E(G)$ and $v \in V(G)$, a cut of the same weight can be obtained by removing ev from $\mathcal{C}$ to be replaced by $se \in E(B)$ instead. Hence, $\mathcal{C}$ can be supposed to only contain edges incident to s or t. Of course, as $uv = e \in E(G)$ has only two outneighbors in B, all of se, ut, vt can not be in $\mathcal{C}$ at the same time as se is then useless in the cut. the information encoded by $\mathcal{C}$ lies in a set of vertices $H \subseteq V(G)$ corresponding to the elements $v \in V(G)$ such that $vt \in \mathcal{S}$. The edges $se \in \mathcal{C}$ then correspond to all the edges from G having at most one endpoint in H.

The value of this cut is precisely equal to $(|E(G)| - |E(G[H])|) \times 1 + |H| \times c$. Hence, we can send $|E(G)|$ units of flow from s to t whenever this cut is larger than $|E(G)| \times 1$, which is possible if and only if any induced subgraph $H \subseteq G$ satisfies $|H| \times c \geq E(H)$, and in other words if and only if $c \geq 2mad(G)$. From this flow problem we obtain a Linear Program minimizing the value $2c$, thus computing $mad(G)$.

Maximum average degree

- Minimize : $2c$

- Such that :
  - A vertex can absorb a charge of at most c

$$\forall v \in V(G), \sum_{\substack{e \in E(G) \\ e \sim v}} x_{e,v} \leq c$$

  - Each edge sends a flow of 1 to its endpoints.

$$\forall e = uv \in E(G), x_{e,u} + x_{e,v} = 1$$

- $x_{e,v}$ is a real positive variable, representing the flow sent by an edge e to one of its endpoints v.

[10] When $c = \lceil mad(G)/2 \rceil$, the st-flow problem in this graph corresponds to the computation of an orientation as presented previously.

Equivalently, the mad can be computed through the dual.

Maximum average degree (dual)

- Maximize : $\sum_{e \in E(G)} x_e$
- Such that :
  - The charges of the vertices sum up to at most 2

$$\sum_{v \in V(G)} x_v \geq 2$$

  - The charge of an edge it a most the charge of each of its endpoints

$$\forall uv = e \in E(G)$$

$$x_e \leq x_v \text{ and } x_e \leq x_u$$

- x_v, x_e are real positive variables

This linear formulation of the maximum average degree is also of interest to define other Mixed Integer Linear Programs involving properties of connectivity or acyclicity (see p.78).

0.5.2 Girth

It is also very common in graph theory to consider another parameter whose information yields a lot of locality, and has been known to simplify usually hard problems. The *girth* of a graph G is defined as the largest integer $g(G)$ such that G does not contain any cycle of length $< g(G)$. Besides, this parameter can be computed by running several breadth-first-searches in a graph, which makes it quite easy to implement.

Knowing the girth of a graph has immediate *local* consequences. When $g(G) > 3$, the neighborhood of a vertex is an independent set – which is just triangle-freeness. In general, it means that the neighborhood at distance $\lfloor g(G)/2 \rfloor$ of a vertex induces is a tree in G.

In planar graphs, knowing the value of the mad actually gives an information on the girth through the inequality $mad(G) < 2 + \frac{4}{g-2}$ which is a consequence of Euler's formula *"#vertices - #edges + #faces = 2"*. Hence, the maximum average degree of a planar graph (and in turn its degeneracy δ^*) is strictly less than 6, as seen in this inequality by substituting g with 3

0.5.3 Discharging

The discharging method is one of the classical tools in the study of planar graphs. It is – most of the time – based on purely local considerations, and aims at proving the existence of specific configurations in a graph. More formally, it is in particular used to prove results of the following shape :

Let $\mathcal{F} = \{F_1, \ldots, F_k\}$ be a fixed (and carefully chosen) family of graphs. Any nonempty graph G of a class $\mathcal{G}$ contains as a subgraph at least one element of $\mathcal{F}$.

Those are often the key lemmas of proofs based on induction. Indeed, in order to prove a result on a class of graphs, it is sometimes possible to ensure that a minimum counter-example could not contain some specific patterns for combinatorial reasons. Obtaining the proof that *any* nonempty graph necessarily contains one of these patterns is the finishing blow asserting that the only admissible counterexample would be the empty graph – which usually is not – hence proving the conjecture.

Though this framework is heavily used on planar graphs, the results it produces can sometimes be generalized to all graphs of bounded maximum average degree (and when this bound is at least 6, it means the result holds on all planar graphs). It can also be used to obtain results on planar graphs of large girth (hence of low average degree) through the formula presented in section 0.5.2.

Let us cite as an illustration – beside the proof of the famous 4 colors theorem [18, 136] – a useful lemma whose proof rests on the discharging method. The first one is a result of van den Heuvel and McGuiness [149], which ensures that there is in any planar graph a vertex of small degree around which at most two neighbors have large degree.

Lemma (van den Heuvel and McGuiness)**.** *Let G be a planar graph with minimum degree at least two. Then there exists a vertex v in G with exactly $d(v) = k$ neighbors $v_1, v_2, \ldots, v_k$ with $d(v_1) \leq d(v_2) \leq \ldots \leq d(v_k)$ such that at least one of the following is true:*

(A1) $k = 2$,

(A2) $k = 3$ *and* $d(v_1) \leq 11$,

(A3) $k = 4$ *and* $d(v_1) \leq 7$, $d(v_2) \leq 11$,

(A4) $k = 5$ *and* $d(v_1) \leq 6$, $d(v_2) \leq 7$, $d(v_3) \leq 11$.

For an concise illustration of discharging to edge-choosability of planar graphs, the reader can refer to the Appendix p.125, where [51] is presented. The same method is used in [53] (see Appendix p.110), in which the case analysis is comparatively much heavier.

Chapter 1

Acyclicity and colorings of planar graphs

1.1 Proper coloring

The main focus of this PhD Thesis is on colorings of planar graphs, on which the classical coloring problems have now long been settled. Let us quickly survey the main related results.

First, one can not avoid mentioning one of the main results in graph theory, and one which has no other subject than colorings on planar graphs.

Theorem (Appel, Hakken – Four colors theorem [18, 136])**.** *The chromatic number of a planar graph is at most four.*

In 1993, Voigt [154] gave an example of a non-4-choosable planar graph, her paper being followed one year later by Thomassen's proof [147] that $ch(G) \leq 5$ for planar graph.

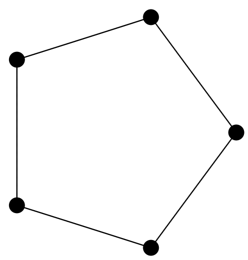
Edge coloring planar graphs is also slowly going out-of-fashion, as researchers eventually answered all the elementary questions (with one small omission). To the question "How could Vizing's theorem be improved in the context of planar graphs", Vizing himself partially answered [153] by proving in 1965 that $\chi'(G) = \Delta(G)$ when $\Delta(G) \geq 8$. This result was not improved upon until 2001.

Theorem (Sanders and Zhao (2001))**.** *[139] A planar graph G with maximum degree at least 7 is $\Delta(G)$-edge-colourable (class I).*

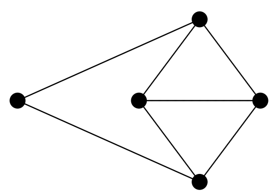
In the same paper, Vizing [153] had provided examples of planar graphs such that $\chi'(G) = \Delta(G) + 1$ for values of $\Delta(G)$ in $\{2, 3, 4, 5\}$ (see Fig.1.1), and conjectured that none existed for larger values. Hence, proper edge coloring of planar graphs still demands some work for the case $\Delta(G) = 6$.

A planar version of the List Coloring Conjecture (see p.29), though, still has to be proved. In 1990, following the same parametrization on the maximum degree, Borodin studied [36] the edge choosability of planar graphs.

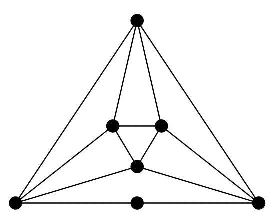
Theorem (Borodin)**.** *If G is a planar graph of maximum degree $\Delta(G) \geq 9$, then G is $(\Delta(G) + 1)$-edge-choosable.*



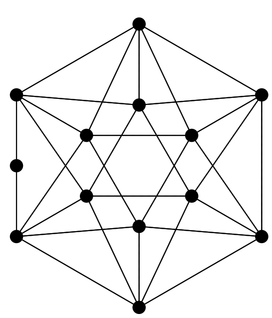
C_5		
$\Delta(G)$	=	2
$\lvert E(G)\rvert$	=	5
Max Matching	=	2
$\chi'(G)$	$\geq$	$5/2 > \Delta(G)$



Subdivided K_4		
$\Delta(G)$	=	3
$\lvert E(G)\rvert$	=	7
Max Matching	=	2
$\chi'(G)$	$\geq$	$7/2 > \Delta(G)$



Subdivided Octahedron		
$\Delta(G)$	=	4
$\lvert E(G)\rvert$	=	13
Max Matching	=	3
$\chi'(G)$	$\geq$	$13/3 > \Delta(G)$



Subdivided Icosahedron		
$\Delta(G)$	=	5
$\lvert E(G)\rvert$	=	31
Max Matching	=	6
$\chi'(G)$	$\geq$	$31/6 > \Delta(G)$

Figure 1.1: Planar graphs with $\chi' = \Delta + 1$ for $\Delta = 2, 3, 4, 5$

Along with Havet, we gave in [51] (see Appendix p.125) a shorter proof of Borodin's result, asserting furthermore that this coloring could be obtained with two elementary operations.

■ How to color the edges of a planar graph with lists of size $\Delta(G) + 1$ when $\Delta(G) \geq 9$ ■

Light edges : Given a graph G, along with an assignment of $\Delta(G) + 1$ colors to each edge, we need not be concerned by any edge $uv \in E(G)$ satisfying $d(u) + d(v) \leq \Delta(G) + 2$. Indeed, if we were to find a feasible coloring of the edges of G from which uv has been removed, we could easily extend this partial coloring by noticing that at most $\Delta(G)$ other edges are incident in G to either u or v, and that as a consequence there exists in the list of uv – of length $\Delta(G)+1$ – at least one color available to color uv.

This also tells us that in the abscence of any light edge, G is of minimum degree three as $d(u) + d(v) \leq \Delta(G) + 2$ trivially holds around vertices of degree one or two. Besides, any vertex of degree three is necessarily adjacent to a vertex of degree $\Delta(G)$.

Light cycles : An even cycle is always 2-choosable (see p.21). If we were to remove from G an even cycle C alternating between vertices of degree 3 and $\Delta(G)$, we would be able, as previously, to color the edges of $G - C$, and then extend the coloring to G. Indeed, in this context any edge of C is incident to at most $\Delta(G) - 1$ edges of $G - C$, and we could, if a feasible coloring of $G - C$ is known, associate to each edge the list of its (at least) two available colors. All is left to do now is to find a feasible assignment for the edges of C.

With Havet, we proved in [51] (see Appendix p.125) that those two operations were actually sufficient to color all planar graphs with $\Delta(G) \geq 9$, using a short discharging argument.

In 1997, along with Kostochka and Woodall [37], Borodin strengthened his result for larger values of $\Delta(G)$ to obtain the best possible version "$ch'(G) = \Delta(G)$".

Theorem (Borodin, Kostochka and Woodall)**.** *If G is a planar graph of maximum degree $\Delta(G) \geq 12$, then $ch'(G) \leq \Delta(G)$*

Besides, the promised edge-coloring can be obtained in linear time thanks to an algorithm of Cole, Kowalik and Škrekovski [55].

1.2 Acyclicity

In the present section, the previous problems of vertex/edge coloring/choosability are once more visited, under the additional constraint of acyclicity. This constraint adds to the definition of proper coloring a global range, which is deadly to most known results on proper coloring :

A proper coloring is said to be *acyclic* if the union of any two colors classes results in an acyclic graph.

Such a kind of coloring ceases to "simply" be a partition of a graph into well-behaved sets, as the sets also need to be well-behaved with each other. Splitting a graph in several parts and building a coloring of the whole graph through the colorings obtained independently can not be achieved anymore without great care and strong hypotheses on the splitting and the color classes themselves.

In the specific context of edge coloring, it generally invalidates proofs based on the use of alternating chains, as reversing them may produce the appearance of many bicoloured cycles which is not easily overcome. It is therefore a tremendous task to generalize result on proper graph coloring – when at all possible.

Acyclic vertex coloring

A rough way to obtain an acyclic coloring of a graph being to ensure that no two vertices of the same color are at distance two, the first upper bound one can get on $\chi_a(G)$ is to apply Brooks' theorem to the square[1] of G, which yields $\chi_(g) \leq \Delta(G)^2 + 1$. Unfortunately, the behavior is quadratic according to the maximum degree while one could have wished for something smaller – indeed, Erdős conjectured [3] in 1976 that one could do much better, namely $\chi_a(G) = o(\Delta(G)^2)$.

This conjecture was finally settled twenty five years later by the following result.

Theorem (Alon, McDiarmid, Reed (1991)[7])**.** $\chi_a(G) = O(\Delta(G)^{4/3})$. *Besides, there exist graphs for which* $\chi_a(G) = \Omega\left(\frac{\Delta(G)^{4/3}}{\log(\Delta(G))^{1/3}}\right)$.

The upper bound, however, is obtained by probabilistic arguments (more precisely by Lovász' Local Lemma, presented in the next section) and does not yield an explicit algorithm (though Lovász' Lemma can be derandomized [120]). This motivates studies on acyclic coloring to fill those gaps, and in 2005 Raspaud and Fertin proved [77] that one could acyclically color a graph with at most $\frac{1}{2}\Delta(G)(\Delta(G) - 1)$ colors in $n\Delta(G)^2$ time.

Acyclic edge coloring

The situation for acyclic edge coloring is not as good, as no tight upper bounds on $\chi'_a(G)$ – the *acyclic chromatic index* of G – are currently known. If some graph G satisfy $\chi_a(G) = \Omega\left(\frac{\Delta(G)^{4/3}}{\log(\Delta(G))^{1/3}}\right)$, none has been witnessed to invalidate the following conjecture, formulated by Fiamčik [78] in 1978 (in Russian) and in 2001 by Alon, Sudakov, and Zaks [10].

Conjecture (Alon, Fiamčik, Sudakov, and Zaks)**.** *For any graph G,* $\chi'_a(G) \leq \Delta(G) + 2$

This conjecture is wide open : if proven true, $K_{3,3}$ and complete graphs of even order would make it tight, but so far any information of the value of $\chi'_a(K_{2n})$ would be a breakthrough. The following asserts that $\chi_a(K_{2n}) \geq \Delta(K_{2n}) + 2$ holds :

Any acyclic edge coloring of K_{2n} can contain at most one perfect matching, as otherwise the union of any two of them contains a cycle. Hence, k color classes can cover at most $n + (k-1)(n-1)$ edges and $\binom{2n}{2}$ edges require at least $2n + 1 = \Delta + 2$ colors.

[1]The *square* of a graph G, written G^2, is the graph on the same vertex set in which two vertices are linked when they are adjacent in G or have a common neighbor.

This is similar to the bound $\chi(G) \geq \frac{|V(G)|}{\alpha(G)}$, and in any case does not prove that a coloring of K_{2n} with $\Delta + 2$ colors exists. The existence of such a coloring would be implied [2] by the following – old, pretty, well-studied – conjecture.

Conjecture (Perfect 1-factorization conjecture (1963))**.** *[110] The edges of the complete graph K_{2n} can be decomposed into $2n - 1$ matchings such that the union of any two of them is a Hamiltonian cycle.*

(Alon notes in [10] that this conjecture has much in common with the other conjecture $\chi_a(K_{2n}) = \Delta(K_{2n}) + 2$, to which it could very well be equivalent)

As with acyclic vertex coloring, the best general upper bound makes use of Lovász' Local Lemma. It is a result of Alon, McDiarmid, and Reed [15] proved in 1991.

Theorem (Alon, McDiarmid, Reed)**.** $\chi'_a(G) \leq 16\Delta(G)$

This remains the best general result today on acyclic edge coloring.

The 16Δ bound and the Local Lemma

The Local Lemma, or Lovász' Local Lemma[3], was first presented in 1975, in a paper from both Erdős and Lovász [69]. If their aim was in this article to prove that k-uniform hypergraphs can be 2-colored [4] when $k \geq 9$, the tool they defined turned out to have an exceptional expressivity : since the original paper, this lemma has been repeatedly used to prove a plethora of existential results, and – among others – the best known general bound on the acyclic chromatic index of a graph [15].

More technically, it is a lemma whose aim is to bound the probability of the simultaneous realization of several *non-independent* probabilistic events. The first moment method (see [9]), illustrated on page 17 to compute the size of a maximum independent set in a random graph $\mathcal{G}_p^n$, already achieved this goal. Given a list of n – possibly dependent – probabilistic events $A_1, \ldots, A_n$, one can roughly bound the probability that none of them is realized by noticing that in the worst case of total dependence, $P[\overline{A_1} \wedge \cdots \wedge \overline{A_n}] \geq 1 - P[A_1] - \cdots - P[A_n]$. This was sufficient in our former situation, but while this inequality can very possibly be an equality from time to time, Lovász' Local Lemma is a tool that can strengthen it by feeding on the information available on the independence between the probabilistic events at hand.

God willing, it all begins with a graph.

Given a set of probabilistic events $\mathcal{A} = \{A_1, \ldots, A_n\}$, one can define on them a (non-unique) dependency graph $G_{\mathcal{A}}$, by defining the adjacency relationship in such a way that non-adjacent events are independent. More is actually required. In the dependency graph $G_{\mathcal{A}}$, Lovász' Local Lemma requires that $A_i \in V(G_{\mathcal{A}})$ be *mutually independent* of all its non-neighbors.

[2] In [10] the authors note that one can obtain an acyclic edge coloring of K_{2n-1} and K_{2n-2} by producing a perfect 1-factorization of K_{2n} and deleting one or two vertices – which breaks any bicoloured cycle.

[3] Unfortunately, László Lovász's productivity sometimes works against himself. Lovász' Local Lemma could have been referred to by the acronym LLL, though this already points to the Lenstra–Lenstra–Lovász algorithm, which computes a "nearly-orthogonal" basis of a discrete vector subspace of $\mathbb{R}^n$.

[4] A 2-coloring of an hypergraph is an assignation of boolean values to its vertices such that no edge is monochromatic – a possible generalization of proper graph coloring.

Definition. An event A is said to be mutually independent from events $B_1, \dots, B_n$ when the probability of A does not depend on any simultaneous information on all – or a subset of – the events $B_1, \dots, B_n$.

This graph contains some information of interest to us. If it is an independent set – meaning that all the events are mutually independent – the probability that none of them occurs is equal to $P[\overline{A_1} \wedge \dots \wedge \overline{A_n}] = P[\overline{A_1}] \times \dots \times P[\overline{A_n}]$.

If the graph is a cycle and $n = 2k$, one can infer that

$$P[\overline{A_1} \wedge \dots \overline{A_{2i+1}} \wedge \dots \wedge \overline{A_{2k-1}}] = P[\overline{A_1}] \times \dots \times P[\overline{A_{2i+1}}] \times \dots \times P[\overline{A_{2k-1}}]$$

and

$$P[\overline{A_2} \wedge \dots \overline{A_{2i+2}} \wedge \dots \wedge \overline{A_{2k}}] = P[\overline{A_2}] \times \dots \times P[\overline{A_{2i+2}}] \times \dots \times P[\overline{A_{2k}}]$$

by arguing that the sets $(A_i)_{i \text{ even}}$ and $(A_i)_{i \text{ odd}}$ are both independent in $G_{\mathcal{A}}$, and hence that the events themselves are independents. From these inequalities, one can deduce that

$$\begin{aligned} P[\overline{A_1} \wedge \dots \wedge \overline{A_n}] \geq & 1 - P[\overline{A_1} \wedge \dots \overline{A_{2i+1}} \wedge \dots \wedge \overline{A_{2k-1}}] - P[\overline{A_2} \wedge \dots \overline{A_{2i+2}} \wedge \dots \wedge \overline{A_{2k}}] \\ \geq & 1 - P[\overline{A_1}] \times \dots \times P[\overline{A_{2i+1}}] \times \dots \times P[\overline{A_{2k-1}}] \\ & - P[\overline{A_2}] \times \dots \times P[\overline{A_{2i+2}}] \times \dots \times P[\overline{A_{2k}}] \end{aligned}$$

which is already much stronger than

$$P[\overline{A_1} \wedge \dots \wedge \overline{A_n}] \geq 1 - P[A_1] - \dots - P[A_n]$$

Of course, if $G_{\mathcal{A}}$ is a complete graph, its information only is not sufficient to improve this upper bound.

The Local Lemma, from a set of events $\mathcal{A}$ and a dependency graph $G_{\mathcal{A}}$, defines a collection of equations from which one can prove $P[\overline{A_1} \wedge \dots \wedge \overline{A_n}] > 0$, hence the existence of such a situation.

Lemma (Lovász Local Lemma)**.** *Let $\mathcal{A} = \{A_1, \dots, A_n\}$ be a set of probabilistic events and $G_{\mathcal{A}}$ a dependency graph for this family. If there exists values $0 \leq x_1, \dots, x_n < 1$ such that the equations*

$$P[A_i] \leq x_i \prod_{A_j A_i \in G_{\mathcal{A}}} (1 - x_j)$$

hold for any i, then $P[\overline{A_1} \wedge \dots \wedge \overline{A_n}] > 0$.

This system of equations can be simplified to yield the symmetric Local Lemma

Lemma (Symmetric Local Lemma)**.** *Let $p = \max_i P[A_i]$ and $d = \Delta(G_{\mathcal{A}})$.*
If $epd \leq 1$, then $P[\overline{A_1} \wedge \dots \wedge \overline{A_n}] > 0$.

In their paper [15], Alon McDiarmid, and Reed give the following result.

Theorem (Alon, McDiarmid, Reed). *For any graph G, $\chi'_a(G) \leq 16\Delta(G)$*

Proof. Let G be a graph, and let its edges be colored independently and uniformly, using colors from a set of $16\Delta(G)$ colors. The events we would like to avoid are the following :

- $A_{e,e'}$ is the event that two incident edges e, e' are colored with the same color. $P[A_{e,e'}] = \frac{1}{16\Delta(G)}$.
- B_C is the event that an even cycle C is properly 2-colored. $P[B_C] \leq (\frac{1}{16\Delta(G)})^{|C|-2}$

No edge e lies in more than $\Delta(G)^{|C|-2}$ cycles of size $|C|$. For any even cycle C, the event B_C is mutually independent from all but $2|C|\Delta(G)$ events of type A, and at most $|C|\Delta(G)^{|C'|-2}$ events of type B_* where $*$ denotes a cycle of size $|C'|$. The values $x_{A_{e,e'}} = \frac{2}{16\Delta(G)}$ and $x_{B_C} = (\frac{2}{16\Delta(G)})^{|C|-2}$ satisfy the equations from Lovász' Local lemma, and prove the result. □

Actual computation

Working on acyclic colorings invariably implies needing – from time to time – to color graphs, whether to prove a result or to help define the most sensible conjecture. When the graphs are becoming less and less trivial, it can be useful to ask for the computer's help, as the rules of acyclic coloring make it tedious to ensure a coloring is valid.

Through the definition of a LP computing the maximum average degree of a graph, it is possible to ensure from the inside of a LP that a set of edges forms a forest (as it is a graph with $mad < 2$), and from there to ensure a set of edges is connected with a small number of constraints.

This method let me define MILP in order to compute optimal acyclic colorings, by combining the classical formulation of vertex/edge coloring, and adding the (costly) constraints that any two of the $\binom{k}{2}$ pairs of color classes induces an acyclic subgraph.

This practical work is presented in Ch.5.1. Note that the same technique can also be used to define LP for other known optimization problems (minor testing, TSP, and others).

1.2.1 Linear arboricity

The linear arboricity of a graph is a parameter whose behavior is quite close to acyclic edge coloring. The arboricity of a graph being the smallest number of edge-disjoint forests required to cover all of its edges, the *linear* arboricity – noted $la(G)$ – denotes the smallest integer k such that the edges of G can be partitioned into k edge-disjoint forests of paths (forests of maximum degree 2).

While any partition of $E(G)$ into k edge-disjoint linear forests results in an proper edge-coloring of G with $2k$ colors (it suffices to properly color the edges of the first color class with 1 and 2, those of the second forest with 3 and 4, etc ...), the converse is not true : an edge-coloring of G does not immediately give a partition into linear forests of $E(G)$, as the union of two color classes probably induces cycles.

Any upper bound on the acyclic chromatic index, however, also bounds the linear arboricity. Indeed, one can obtain a covering of the edges of a graph into linear forests by grouping the color classes of an acyclic edge coloring into pairs ($\{\{1,2\},\{3,4\},\{5,6\},\dots\}$), each of them inducing an acyclic graph of maximum degree 2, yielding the inequality $la(G) \leq \lceil \frac{1}{2}\chi'_a(G) \rceil$.

The converse, on the other hand, is not true : the proper edge coloring produced by 2-coloring the edges of each linear forest has no reason to be acyclic. Indeed, while colors 1 and 2 induce an acyclic graph there is no reason why such a property should hold between colors 1 and 3. The comparative strength of the best bound on the linear arboricity, over the best bound on acyclic edge coloring has to lie in the fact that only k pairs of colors are required to induce an acyclic graph, instead of $\binom{k}{2}$ in the latter case.

As a linear forest can contain at most two edges incident to the same vertex, the inequality $la(G) \geq \lceil \frac{1}{2}\Delta \rceil$ holds in general, but acyclicity being hard to handle the upper bounds on this number are considerably loose.

In 1980, Akiyama, Exoo, and Harary [2] conjectured a situation echoing of edge coloring.

Conjecture (Akiyama, Exoo, Harary)**.** *For any graph G,* $\lceil \frac{1}{2}\Delta(G) \rceil \leq la(G) \leq \lceil \frac{1}{2}(\Delta(G)+1) \rceil$

Such a conjecture would be tight. However, there is still between this conjecture and the best general upper bound obtained so far – a result of Alon, Teague, and Wormald [12] in 2001 – a polynomial difference.

Theorem (Alon, Teague, and Wormald (2001) [12])**.** *There is an absolute constant c such that any graph G of maximum degree Δ satisfies*

$$la(G) \leq \frac{\Delta}{2} + c\Delta^{2/3}(\log \Delta)^{1/3}$$

It is to be noted that similarly to the 16Δ bound on χ'_a, this upper bound was obtained through Lovász' Local Lemma (see p.41).

We may be in need of a class of graph achieving higher values of la, or better tools to improve the upper bounds, but there is definitely room for studies in this area. The structure of planar graphs, however, is simple enough to balance our misunderstandings of the general case.

Theorem (Cygan, Hou, Kowalik, Lužar, Wu [58])**.** *The linear arboricity conjecture holds for planar graphs G with* $\Delta(G) \geq 10$.

The intersection of three matroids

The activity of covering – or partitioning – the edges of a graph into forests of paths is a sport that can only become fashionable once several of its relaxations are deemed understood, and it is a fact that it is actually the meeting point of several more classical problems.

Indeed, partitioning a graph into linear forests is first trying to partition its edges into arbitrary forests. The problem of determining the minimum number of such forests – or actually computing them – has been addressed with a complete characterization from Nash-Williams [124] as early as in 1964.

Theorem (Nash-Williams)**.** *The smallest number of forests necessary to cover the edges of a graph G is equal to*

$$\max_{H \subseteq G} \left\lceil \frac{|E(H)|}{|V(H)|-1} \right\rceil$$

This result is a complete characterization of graphs that can be written as the disjoint union of k forests, as for any induced subgraph $H \subseteq_i G$ the disjoint union of k forests can cover at most $k(|V(H)|-1)$ edges.

Hence this theorem from Nash-Williams addresses the relaxed version of linear arboricity in which degree constraints are left out, but one could have attempted to take a different direction and relax the acyclicity constraint. At the end of this road lies one of Petersen's theorems [129].

Theorem (Petersen)**.** *A $2k$-regular graph is the union of k edge-disjoint 2-regular graphs.*

The promised decomposition – and a proof of this theorem – can be obtained in two steps, if one is willing to work on directed graphs which (for once) greatly simplify the problem.

■ How to decompose the edges of a graph into edge-disjoint graphs of maximum degree 2 ■

---

1. Find an orientation D of G of maximum out-degree $\lceil \frac{1}{2} mad(G) \rceil \leq \lceil \frac{1}{2} \Delta \rceil$ (see p.33). If G is $2k$-regular, then the maximum out-degree $\Delta^+(D)$ is equal to k.

2. Build a bipartite graph B having on side L a vertex v^- and on side R a vertex v^+ for any vertex v of G.

3. For each edge $uv \in D$, link u^- to v^+ in B. If G is $2k$-regular, then B is r-regular.

4. Compute a proper edge coloring of B, which can be done with at most $\lceil \frac{1}{2} \Delta \rceil$ color classes.

5. The partition of $E(G)$ is obtained by associating each edge to its corresponding color class in B. Each vertex v has at most two neighbors of the same class (one for v^-, one for v^+), with a total of at most $\lceil \frac{1}{2} \Delta \rceil$ different colors (k if G is $2k$-regular).

It is hence possible to partition any graph in edge-disjoint graphs of maximum degree two, while such a construction may give rise to many cycles and no clear indication on how to prevent it.

This proof, however, is clear enough about the interests of dealing with a directed version of our problem. And indeed, this question is also one of the most disturbing conjectures related to linear arboricity.

Conjecture (Nakayama, Peroche [122])**.** *The arcs of a directed graph D can be covered with at most $\Delta^+(D) + 1$ forests of directed paths.*

Were this conjecture true, one could obtain a decomposition of the edges of a graph G into linear forest by working on an orientation of G minimizing the out-degree (see p.33).

From this conjecture on directed graphs, one can think of linear arboricity as the intersection of three matroids :

- A first matroid whose bases are the sets of arcs inducing a digraph of maximum out-degree 1
- A second matroid whose bases are the sets of arcs inducing a digraph of maximum in-degree 1
- A third matroid whose bases are the sets of arcs inducing (when forgetting their orientation) a forest (the cycle matroid).

The problems of covering the edges of a graph with bases from only one of those matroids can be solved using the Matroid Union Theorem (see Schrijver [143]) – the third one actually leads directly to Nash-Williams' Theorem. Covering the edges of a graph with elements from any two of these matroids yields Petersen's theorem when the first two are chosen. There exists, however, no Union Theorem for the intersection of two matroids though it is computationally easy to find a maximal base of both matroids. Even finding such a maximal base for the intersection of 3 matroids, though, is already NP-Hard. Being able to find a maximal common base of these three matroids would mean being able to decide whether a graph admits a Hamiltonian path, and that may be asking for too much in polynomial time.

1.2.2 Acyclic edge coloring of planar graphs

As it is the case with the linear arboricity, many hard conjectures virtually disappear when studied in the context of planar graphs. Instead of having to deal with a state-of-the-art bound of 16Δ on the acyclic chromatic index of a graph, several stronger results were known.

Theorem (Fiedorowicz, Haluszczak, Narayanan [79])**.** $\chi'_a(G) \leq 2\Delta(G) + 29$ *for any planar graph* G*.*

Or, when restricted to triangle-free graphs :

Theorem (Hou, Liu, Wang (2010) [105])**.** *Let G be a triangle-free planar graph with maximum degree $\Delta(G) \geq 8$. Then $\chi'_a(G) \leq \Delta(G) + 3$.*

Theorem (Yu, Hou, Liu, Liu, Xu (2009) [157])**.** *Let G be a triangle-free planar graph with maximum degree $\Delta(G) \geq 12$. Then $\chi'_a(G) = \Delta(G)$.*

Both papers [105, 157] also studied the acyclic chromatic index of planar graphs with given girth and maximum degree.

With Basavaraju, Chandran, Havet and Müller, we divided the general bound for planar graphs by two [22] (see Appendix p.91), obtaining the exact asymptotic behavior.

Theorem (Basavaraju, Chandran, Cohen, Havet and Müller)**.** *If G is a planar graph*

$$\chi'_a(G) \leq \Delta(G) + 12$$

Our proof heavily rests on the theorem from van den Heuvel and McGuiness presented on page 36. We also conjectured that the additive term could be removed for large values of $\Delta(G)$, and proved that Δ colors were sufficient for graphs of large girth.

Theorem (Basavaraju, Chandran, Cohen, Havet and Müller)**.** *If G is a planar graph with girth at least 5, then $\chi'_a(G) = \Delta(G)$.*

It is also to be noted that on the class of outerplanar graphs – taken as a subclass of 2-degenerate graphs – a tight result has been obtained.

Theorem (Basavaraju, Sunil Chandran (2010))**.** *[21] Let G be a 2-degenerate graph. Then $\chi'_a(G) \leq \Delta(G) + 1$*

1.2.3 Linear and 2-frugal choosability in graphs of small maximum average degree

Along with Havet [53] (see Appendix p.110), I studied two variants of proper vertex coloring in the context of sparse graphs.

The first one is 2-frugal coloring, which is an acyclic coloring deprived of its global constraints. It is actually a proper coloring of the vertices of a graph in which each pair of color classes induces a graph of maximum degree 2, and so can be defined by purely local constraints. We also studied the non-local version – linear coloring – in which pairs of color classes must also induce acyclic graphs.

These global constraints complicate our discharging arguments, which were applied to the choosability version of these colorings of graphs with small maximum average degree, generalizing results obtained by Esperet, Montassier, and Raspaud [73] on planar graphs.

About these two colorings, we obtained through the discharging method sufficient conditions on the mad and maximum degree (or alternatively girth and maximum degree) of planar graphs ensuring the existence of colorings with a small number of colors.

$mad(G)$	$\Delta(G)$	Linear choosability	
$< \frac{16}{7} \approx 2.2857$	≥ 3	$\leq \lceil \frac{\Delta}{2} \rceil + 1$	Esperet et al. [74]
$< \frac{39}{16} = 2.4375$	≥ 5	$\leq \lceil \frac{\Delta}{2} \rceil + 1$	Appendix p.110
$< \frac{48}{19} \approx 2.5263$	≥ 7	$\leq \lceil \frac{\Delta}{2} \rceil + 1$	Appendix p.110
$< 3 - \frac{3}{\Delta+1}$	≥ 8	$\leq \lceil \frac{\Delta}{2} \rceil + 1$	Appendix p.110
$< \frac{5}{2}$		$\leq \lceil \frac{\Delta}{2} \rceil + 2$	Esperet et al. [74]
$< \frac{60}{23} \approx 2.6086$	≥ 5	$\leq \lceil \frac{\Delta}{2} \rceil + 2$	Appendix p.110
$< 3 - \frac{9}{4\Delta+3}$	≥ 7	$\leq \lceil \frac{\Delta}{2} \rceil + 2$	Appendix p.110
$< \frac{14}{5} = 2.8$		$\leq \lceil \frac{\Delta}{2} \rceil + 3$	Appendix p.110
< 3	≥ 12	$\leq \lceil \frac{\Delta}{2} \rceil + 3$	Appendix p.110
< 3		$\leq \lceil \frac{\Delta}{2} \rceil + 4$	Appendix p.110
$mad(G)$	$\Delta(G)$	2-frugal choosability	
$< \frac{5}{2}$	≥ 7	$\leq \lceil \frac{\Delta}{2} \rceil + 1$	Appendix p.110
< 3		$\leq \lceil \frac{\Delta}{2} \rceil + 3$	Appendix p.110

Or alternatively, according to the girth of planar graphs :

girth	$\Delta(G)$	Linear choosability	
≥ 16	≥ 3	$\leq \lceil \frac{\Delta}{2} \rceil + 1$	Esperet et al. [74]
≥ 7	≥ 13	$\leq \lceil \frac{\Delta}{2} \rceil + 1$	Raspaud and Wang [133]
≥ 8		$\leq \lceil \frac{\Delta}{2} \rceil + 1$	Appendix p.110
≥ 10		$\leq \lceil \frac{\Delta}{2} \rceil + 2$	Esperet et al. [74]
≥ 9	≥ 5	$\leq \lceil \frac{\Delta}{2} \rceil + 2$	Appendix p.110
≥ 7		$\leq \lceil \frac{\Delta}{2} \rceil + 3$	Appendix p.110
≥ 6		$\leq \lceil \frac{\Delta}{2} \rceil + 4$	Appendix p.110
≥ 5		$\leq \lceil \frac{\Delta}{2} \rceil + 6$	Raspaud and Wang [132]
	≥ 85	$\leq \lceil \frac{9}{10}\Delta \rceil + 5$	Raspaud and Wang [132]
girth	$\Delta(G)$	2-frugal choosability	
≥ 6		$\leq \lceil \frac{\Delta}{2} \rceil + 3$	Appendix p.110

Chapter 2

Graph and Hypergraph decomposition

In this chapter are presented two decomposition problems, in the settings of graphs and hypergraphs. First of all, let us say that a *hypergraph* $\mathcal{H}$ (or *set system*) defined on a set X of vertices is nothing but a family of subsets of X. Hence, saying that $\mathcal{H} \subseteq 2^X$ is by itself a definition of the object, like one could define a graph G to be a subset of $\binom{X}{2}$.

A hypergraph can be considered to be a generalization of graphs, like graphs can be considered to be a generalization of boolean logic. If much of the terminology is common to graphs and hypergraphs (one will talk about vertices in both situations, edges are renamed as hyperedges, the degree of a vertex in a hypergraph is the number of hyperedges containing this vertex and regularity has the same definition), few are the graph problems which remain meaningful when defined in the setting of hypergraphs. In particular, several possible generalizations are available for the most basic graph notions of path, cycle, connectivity, or for proper colorings.

One also has to consider the numerical difference between these combinatorial objects, i.e. the numerical difference between the number of hypergraphs on n elements, 2^{2^n}, and the number of graphs on n elements, $2^{\binom{n}{2}}$. This is a hint that the average hypergraph encodes an amount of information way above what a graph on the same amount of vertices (or a polynomial number of them) contains, and that for this reason most of the hypergraphs *produced* by a graph problem will inherit a very constrained structure, unrepresentative of the behavior of "general hypergraphs".

Hypergraphs, when studied for themselves, have given rise to very elegant results. One could for instance name Baranyai's theorem (whose proof rests on computations of flows and/or matchings, see p.15), or the complete characterization of the profile of Sperner families extending the LYM inequality (see [30]). However, the most enticing aspect of hypergraph theory is definitely its list of open problems, if only for Frankl's conjecture on frequent vertices in union-closed families, or Chvatal's conjecture on maximum intersecting families in hereditary hypergraphs (see p.9 for both). The interested reader can refer to Bollobas [30] or Berge [26] for any additional information.

Besides hypergraphs, the results presented in the current chapter will use one of the fundamental results of design theory, whose beginning can probably be traced back to the following problem.

Kirkman's Problem (1850)

Fifteen schoolgirls walk each day arranged three-by-three in five rows. How can their positions be planned in such a way that during one week no two of them are twice in the same row ?

Kirkman's problem is actually to find, given a set of n schoolgirls (improperly named *vertices* in the whole body of scientific literature), a set of $(n-1)/2$ partitions of $[n]$ by elements of $\binom{[n]}{3}$ such that any two schoolgirls appear in at most one 3-set (and hence *exactly* one for cardinality reasons). When $n = 15$, this problem has a solution (see Fig.2.1).

Day 1	$\{X, A, a\}$	$\{E, G, f\}$	$\{C, D, g\}$	$\{B, F, d\}$	$\{b, c, e\}$
Day 2	$\{X, B, b\}$	$\{F, A, g\}$	$\{D, E, a\}$	$\{C, G, e\}$	$\{c, d, f\}$
Day 3	$\{X, C, c\}$	$\{G, B, a\}$	$\{E, F, b\}$	$\{D, A, f\}$	$\{d, e, g\}$
Day 4	$\{X, D, d\}$	$\{A, C, b\}$	$\{F, G, c\}$	$\{E, B, g\}$	$\{e, f, a\}$
Day 5	$\{X, E, e\}$	$\{B, D, c\}$	$\{G, A, d\}$	$\{F, C, a\}$	$\{f, g, b\}$
Day 6	$\{X, F, f\}$	$\{C, E, d\}$	$\{A, B, e\}$	$\{G, D, b\}$	$\{g, a, c\}$
Day 7	$\{X, G, g\}$	$\{D, F, e\}$	$\{B, C, f\}$	$\{A, E, c\}$	$\{a, b, d\}$

Figure 2.1: A solution to Kirkman's problem [43]

Ray-Chaudhuri and Wilson proved [134] in 1971 that such a system exists if and only if $n \equiv 3[6]$. In what will concern us from now on, however, there will be little need for the additional constraint that the collection of 3-sets be partitioned into partitions of $[n]$. What we are aiming at are Steiner Triple Systems.

Definition. A *Steiner Triple System* of order n is a subset S of $\binom{[n]}{3}$ such that each pair of elements in $[n]$ appear in exactly one element of S. Equivalently, a Steiner Triple System is a partition of the edges of the complete graph K_n into triangles.

As previously, such set systems need not necessarily exist for any value of n. Indeed, as each 3-set contains 3 different pairs, and as each pair is required to appear in exactly one 3-set, we already know that such systems can only exist when $\binom{n}{2} \equiv 0[3]$, and so when $n \equiv 0, 1[3]$. Besides, as any vertex appears in two different pairs for any 3-set that contains it, the total number of pairs in which it appears, $n-1$, must be divisible by 2. Those two necessary conditions, summed up by $n \equiv 1, 3[6]$, are actually sufficient.

Theorem (Kirkman (1847))**.** *A Steiner Triple System exists if and only if* $n \equiv 1, 3[6]$.

Since, this theorem has been powerfully generalized twice by Wilson, producing the highly regular structures which are typical of design theory. If his theorems remain (heavily) asymptotic – it actually proves that the necessary constraints are also sufficient for large enough values of n – exact constructions are actually available [54] when $k \leq 9$.

Theorem (Wilson (1975))**.** *For any fixed k and any large enough integer n, the edges of K_n can be partitioned into edge-disjoint copies of K_k if and only if $k-1$ divides $n-1$ and $\binom{k}{2}$ divides $\binom{n}{2}$.*

Wilson's second generalization involves decomposing the edges of K_n into copies of an arbitrary graph H, and in this case the necessary conditions are to be changed as H is not necessarily regular anymore. His answer asserts that once more an arithmetical answer is sufficient : if a vertex of K_n belongs to different edge-disjoint copies of H, it may successively represent different vertices of H, and in particular its degree in K_n is the sum of the degrees of the vertices it represents. Hence, there exists an integer linear combination using the elements of H's degree sequence which produces $\Delta(K_n) = n-1$, which in turn is necessarily divisible by the greatest common divisor of the elements of H's degree sequence.

Theorem (Wilson (1975))**.** *For any fixed graph H and any large enough integer n, the edges of K_n can be partitioned into edge-disjoint copies of H if and only if $gcd(d_H(v_1), \dots, d_H(v_{|V(H)|}))$ divides $n-1$ and $|E(H)|$ divides $\binom{n}{2}$.*

2.1 Arboricity in hypergraphs

Because there exist in hypergraphs multiple definitions of paths and cycles, there exist as well many different definitions of what an acyclic hypergraph is. As we would like an acyclic hypergraph to generalize acyclic graphs, we could attempt to copy a definition of what a forest is.

"A forest is a graph G verifying the equality $|V(G)| = |E(G)| + cc(G)$."
(where $cc(G)$ denotes the number of connected components of G)

This, however, would require the definition of *connectedness* in hypergraphs. Similarly, we could attempt to generalize this definition of a tree.

"A tree is a graph such that exactly one path exists between any two vertices."

But once more we should then define what a path is in hypergraphs. In the following, the definition of acyclicity we study is inherited from this alternative definition of a forest.

"A forest is a graph that can be built by starting from one vertex and repeatedly adding vertices of degree one."

A hypergraph $\mathcal{H}$ (defined over the set X of vertices) is said to be *α-acyclic*[1] when Graham's reduction algorithm [88] – i.e. the repetition of the following rules – reduces $\mathcal{H}$ to the empty hypergraph.

- If a vertex $x \in X$ has degree one, then delete x from the edge containing it.
- If $A, B \in \mathcal{H}$ are distinct edges such that $A \subseteq B$, then delete A from $\mathcal{H}$.

With these rules, one can say that a k-uniform hypergraph (i.e. a hypergraph whose hyperedges all have cardinality k) is α-acyclic if and only if it is 1-degenerate according to this given generalization of degeneracy. As for graphs, researchers studying hypergraphs are interested in possible equivalents of Nash Williams' [124] theorem (see p.45).

With Bermond, Chee, and Zhang [27] (see Appendix p.129) we studied the α-arboricity of the complete 3-uniform hypergraphs, which is the minimum number of α-acyclic hypergraphs necessary to cover (or equivalently to partition) the edges of the complete 3-uniform hypergraph $\binom{X}{3}$.

The starting point on this problem is the observation that an α-acyclic k-uniform hypergraph on n vertices can have at most $n-k+1$ edges. This can actually be deduced from an equivalent definition of the α-acyclicity of hypergraphs based on induced cycles (through chordality[2]).

[1] For alternative definitions of hypergraph acyclicity, namely Berge-acyclicity, β-acyclicity and γ-acyclicity, see [75]

[2] A graph is said to be *chordal* if it has no induced cycle of length ≥ 4, or alternatively if it can be reduced to the empty graph by successive removals of vertices whose neighborhood is a clique.

Definition (k-sections of hypergraphs (see Berge [26])). The *k-section* of a hypergraph $\mathcal{H}$ is the k-uniform hypergraph defined on the same vertex set whose edges are all the k-subsets contained in an edge of $\mathcal{H}$. The 2-section of a hypergraph is a graph.

Theorem (Beeri et al. [23]). *$\mathcal{H}$ is α-acyclic if and only if its 2-section G is a chordal graph whose maximal cliques are the edges of $\mathcal{H}$.*

A result of Acharya and Las Vergnas [1],[27], through the definition of the *cyclomatic number* gives a complete characterization of α-acyclic hypergraphs pleasantly reminiscing of trees.

Definition (α-acyclic uniform hypergraph, alternative definition [23],[1],[27]). A k-uniform hypergraph is α-acyclic if and only if it can be built from one edge by iteratively creating an edge between isolated vertices and $\leq k-1$ vertices which already belonged to a common hyperedge.

Knowing that a k-uniform α-acyclic hypergraph can have at most $n-k+1$ edges, one cannot hope to partition the edges of an hypergraph in fewer than $\lceil |E(\mathcal{H})|/(n-k+1)\rceil$ α-acyclic hypergraphs. In particular, it has been conjectured that this lower bound is tight for k-uniform complete hypergraphs.

Conjecture (Wang [156]). *The α-arboricity of the complete k-uniform hypergraph on n vertices is*

$$\left\lceil \frac{1}{k}\binom{n}{k-1}\right\rceil$$

As it is often the case for such design problems, symmetrical constructions only exist under some arithmetical constraints. In the specific case of the complete 3-uniform hypergraph, the bound becomes $\frac{1}{3}\binom{n}{2} = \frac{n(n-1)}{6}$ and one can then hope to obtain symmetrical constructions when this value needs not to be rounded, namely when $n \equiv 1, 3[6]$.

That this is precisely the existence condition for Steiner triple systems presented previously should come as no surprise.

Indeed, it is possible to find a partition of a 3-uniform hypergraph on $3n'+1$ or $3n'+3$ vertices into *star-shaped* α-acyclic hypergraphs (see Fig.2.2). Those stars are built from a *central* hyperedge $\{u, v, w\}$, to which are added other 3-sets intersecting $\{u, v, w\}$ in two elements. Hence, all the edges of this *star-shaped* hypergraph are of the form $\{u, v, \bullet\}$, $\{u, \bullet, w\}$ or $\{\bullet, v, w\}$. We also require that no vertex except u, v, or w appear in more than one edge of the star.

Such an hypergraph is naturally α-acyclic, according to the definition given previously, as all the vertices different from u, v or w are of degree 1. Through Graham's rules they can be reduced to the *center* only, and from there to the empty hypergraph. Hence, in order to determine the α-arboricity of the complete 3-uniform hypergraph we can attempt to decompose it into $\frac{n(n-1)}{6}$ *star-shaped* hypergraphs.

All is left to do now is to chose the centers of our future $\frac{n(n-1)}{6}$ α-acyclic hypergraphs, and associate to each of them $n-3$ other sets so that all the hypergraphs are of cardinality $n-k+1 = n-2$. This second step can be reduced to solving some maximum matching problem in a bipartite graph G.

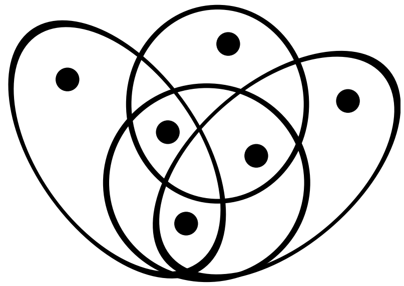


Figure 2.2: A *star-shaped* 3-uniform hypergraph

From a decomposition problem to a perfect matching

On the first side A appear in this graph vertices representing the $\frac{n(n-1)}{6}$ centers we chosed, and on the second side B all the other 3-sets of $\binom{n}{3}$.

We aim at associating many 3-sets to each center, and we would like to ensure that any vertex not contained in the center of the hypergraph is of degree 1. Therefore, we duplicate each of the vertices representing the centers $n-3$ times, and *specialize* them : we actually replace each center $\{u, v, w\}$ by $n-3$ copies $(\{u, v, w\}, t)$ for each vertex $t \notin \{u, v, w\}$, and link $(\{u, v, w\}, t)$ to any 3-set of side B containing t and two vertices among $\{u, v, w\}$, i.e. the possible *extensions* of $\{u, v, w\}$ containing t.

See Fig.2.3 for the case $n = 7$.

If there exists in this graph a perfect matching our problem is solved. Indeed, a perfect matching means that each center $\{u, v, w\}$ is associated to $n-3$ sets of size 3 which are the neighbors of the vertices of the shape $(\{u, v, w\}, t)$ and would yield the desired partition. This being said, a perfect matching in this graph need not necessarily exist. In particular, a vertex $(\{u, v, w\}, t)$ could be isolated if we picked all of $\{u, v, t\}, \{u, t, w\}$ and $\{t, v, w\}$ as centers, and ensuring that there is none of the blocking configuration described by Hall's theorem does not sound like an easy task either.

Hopefully, we can prove the existence of a perfect matching using a simpler criterion by taking as centers the elements of a Steiner Triple System. We are sure in this case that none of $\{u, v, t\}, \{u, t, w\}$ or $\{t, v, w\}$ is a center if $\{u, v, w\}$ is one, and so all the vertices of A are of degree 3. If we note STS_{uv} the unique set from our Steiner Triple System containing both u and v, it is also true to say that any vertex $\{u', v', w'\}$ of B is linked to the three elements $(S_{uv}, w), (S_{uw}, v)$ and (S_{vw}, u), and G is actually both bipartite and 3-regular.

For these reasons the graph admits a perfect matching, and the complete hypergraph on $3n' + 1$ or $3n' + 3$ vertices admits a partition into $\frac{n(n-1)}{6}$ α-acyclic hypergraphs.

With Bermond, Chee, and Zhang [27], we used this construction to completely determine the α-arboricity of the 3-uniform complete hypergraphs.

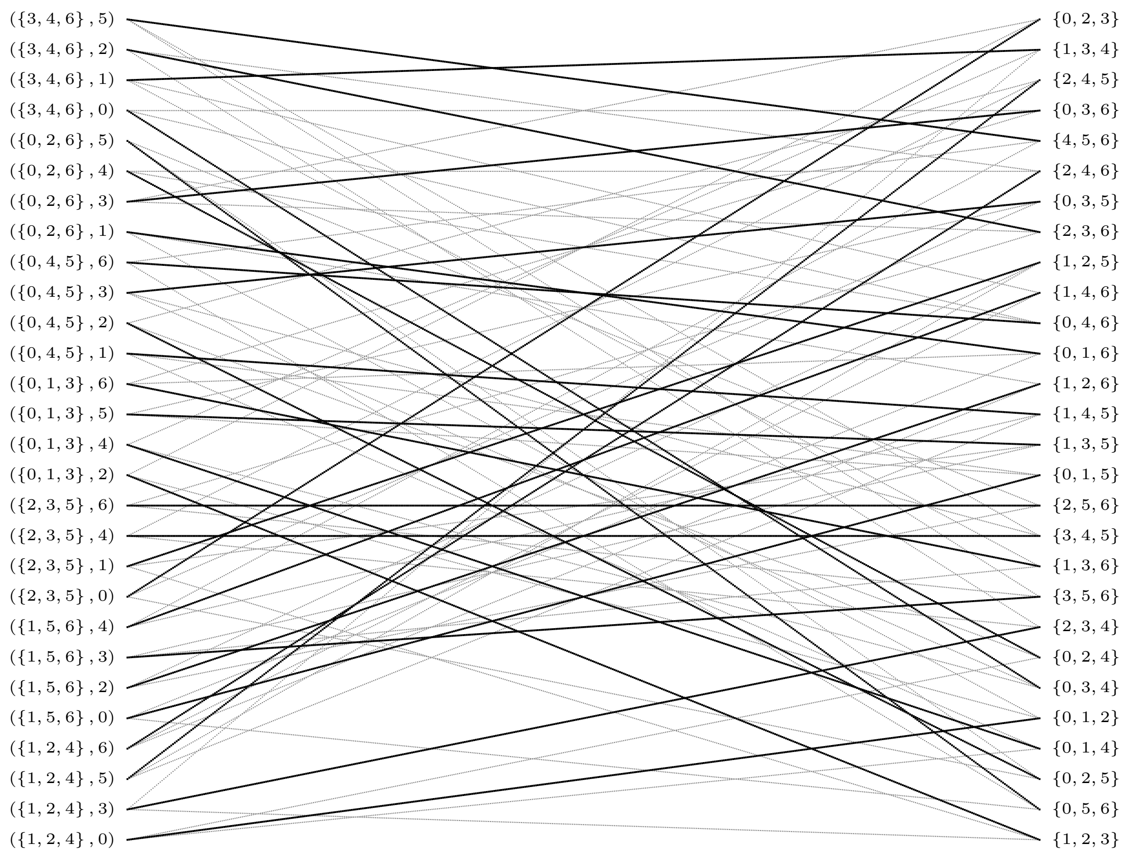


Figure 2.3: Case $n = 7$

Theorem (Bermond, Chee, Cohen, Zhang [27] (see Appendix p.129))**.** *The α-arboricity of the complete 3-uniform hypergraph is $\lceil n(n-1)/6 \rceil$.*

2.2 Induced decompositions

The results and constructions presented in this section have been obtained in collaboration with Zsolt Tuza.

In 2010, Bondy and Szwarcfiter [33] began the study of *induced decompositions* of graphs. While one common definition of a H-decomposition of a graph G is "a partition of $E(G)$ into copies of H", they were interested in finding a partition of $E(G)$ into *induced* copies of H. Of course, an induced H-decomposition of G is always a H-decomposition of G.

The following C_5-decomposition of K_5 (see Fig.2.4) is obviously not induced, while the C_4-decomposition of the octahedron (see Fig.2.5) is a typical example of the constructions they are interested in.

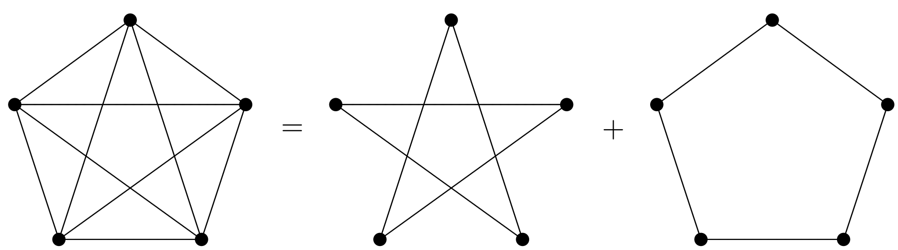


Figure 2.4: Non-induced C_5-decomposition of K_5

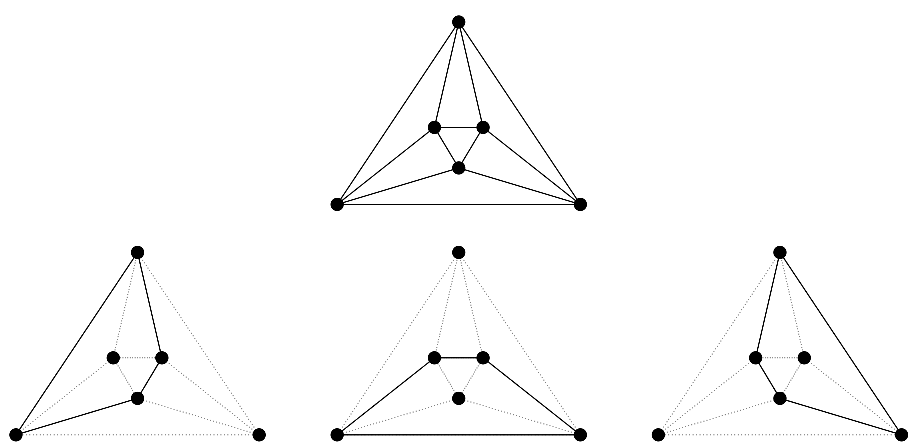


Figure 2.5: Induced C_4-decomposition of the octahedron

They were in particular interested in determining a parameter whose study for usual graph decomposition is completely settled. Indeed, while the most general version of Wilson's theorem (see p.51) asserts that any fixed graph H decomposes the edge set of arbitrarily large complete graphs, there is no way to decompose any complete graph into induced copies of another graph H whenever

H is not complete. They introduced [33] along with their new variant of graph decompositions the parameter $ex(n, H)$ equal to the maximum number of edges of a graph on n vertices admitting an induced H-decomposition.

Then again, by Wilson's Theorem, one can associate to any graph H an infinite sequence of integers n such that K_n admits a H-decomposition. On the other hand, when H is not complete we know for sure that $ex(n, H) < \binom{n}{2}$ for any integer n, which leaves wide open the problem of determining the exact value of ex.

Bondy and Szwarcfiter initiated [33] their study by determining the exact value of $ex(n, H)$ for several families of graphs, most of the effort going in trying to determine the extremal graphs admitting an induced H-decomposition. In particular, they determined the exact value of $ex(n, H)$ when H is the complete r-partite graph with k vertices in each part or the star $K_{1,k}$. They also studied the value of $ex(n, H)$ for small graphs H, getting exact values for C_4, $K_1 + K_2$, $2K_1 + K_2$, $K_1 + K_{1,2}$, and asymptotic values for $K_4\backslash e$. In all these cases the order was $\binom{n}{2} - o(n^2)$. They also gave lower bounds for P_4 (of order $3n^2/8$) and $K_{1,3} + e$ (of order $2n^2/5$).

Short of attempting to determine for all graphs the exact value of $ex(n, H)$, or the corresponding extremal graphs admitting an induced H-decomposition, their results can lead to think that all graphs H satisfy $ex(n, H) = \binom{n}{2} - o(n^2)$. In particular, the smallest graphs for which the asymptotic behavior of ex was unknown after their paper counts $2K_2, P_4, K_{1,3} + e$, to which they added C_5 and C_6.

2.2.1 Induced $2K_2$-decomposition of dense graphs

It turns out that $2K_2$ – which is the disjoint union of two edges – can be used to decompose arbitrarily dense graphs.

Kneser's graph K_k^n is the graph built on all k-subsets $\binom{[n]}{k} \subseteq 2^{[n]}$ of $[n]$, two of them being adjacent whenever they correspond to disjoint sets. For a fixed k, the number of edges in K_k^n is $(1-o(1))\binom{\binom{n}{k}}{2}$, as two random k-subsets of a n-element set are disjoint with high probability when n is grows. Hence, proving that K_k^n admits an induced $2K_2$-decomposition would yield $ex(n, 2K_2) = \binom{n}{2} - o(n^2)$.

This being said, and even though $2K_2$ is always an induced subgraph of a large enough K_k^n with k fixed (a necessary condition for an induced $2K_2$-decomposition to exist), there is no hope of being able to decompose K_2^n into induced copies of $2K_2$, as its number of edges may well be odd at times. The graph K_2^4 is actually isomorphic to $3K_2$. The family K_3^n, however, is more promising as it always has an even number of edges.

Once we are decided on trying to find an induced $2K_2$ decomposition of K_3^n, one way to obtain one presents itself. The first non-empty graph in the family is K_3^6, and it is isomorphic to $10K_2$ – a graph that trivially decomposes into induced $2K_2$. When n is grows large, many copies of K_3^6 can be found in K_3^n : one but needs to consider, for any subset $S \subseteq \binom{[n]}{6}$, the graph induced by the vertices of K_3^n included in S. There will be, for sure, precisely $\binom{6}{3}$ of those, and the graph they induce is without any surprise a copy of K_3^6.

Hence, any *restriction* of K_3^n to a 6-subset S results in an induced copy of $10K_2$. Besides, all these copies are edge-disjoint, as there is only one set of size 6 containing the two disjoint sets of size 3 corresponding to adjacent vertices in K_3^n. One then obtains an induced $2K_2$-decomposition of K_3^n

by considering first an induced K_3^6-decomposition of K_3^n, then splitting the disjoint copies of K_3^6 into induced copies of $2K_2$.

2.2.2 Induced decomposition of dense graphs (second construction)

If the previous technique lets one prove that $ex(n, 2K_2) = \binom{n}{2} - o(n^2)$, obtaining the same result for graphs like P_4 or $K_{1,3} + e$ seems to require additional work. Indeed, if the graphs K_3^7 or K_3^8 can be expected to have a more complicated structure [3] than K_3^6, it is not as easy to decompose a "large" K_3^n into induced copies of K_3^7. Taking a set S of size 7 would indeed lead us toward an induced copy of K_3^7 inside of K_3^n, but all those we obtain this way have no reason to remain edge-disjoint anymore. In order to prove $ex(n, H) = \binom{n}{2} - o(n^2)$ for H equal to P_4 or $K_{1,3} + e$, we need to modify our initial class of graphs to obtain easier decompositions.

Let S_1 and S_2 be two disjoint sets on n_1 and n_2 elements. We define the graph $\mathcal{G}_{n_1,n_2}$ over the elements $S \subseteq S_1 \cup S_2$ such that $|S \cap S_1| = 1$ and $|S \cap S_2| = 1$, two of them being adjacent whenever they are disjoint, as they were in Kneser's graph. This is again a family of dense graphs as the probability that two random sets be disjoint is equal to $(1 - \frac{1}{n_1})(1 - \frac{1}{n_2})$.

(Equivalently, the graph $\mathcal{G}_{n_1,n_2}$ can be described as the complement of the line graph of a complete bipartite graph on $n_1 + n_2$ elements, i.e. $\mathcal{G}_{n_1,n_2} \approx \overline{L(K_{n_1,n_2})}$. This remark will prove useful later on, though we will stick to the terminology of Kneser graphs in what follows.)

The first non-empty graph of this family happens to be 1-regular with 4 vertices... hence $\mathcal{G}_{2,2} \approx 2K_2$. Besides, it is possible as previously to find an induced decompositions of some large $\mathcal{G}_{n,n}$ into copies of a fixed $\mathcal{G}_{2,2}$. Indeed, an induced copy of $\mathcal{G}_{2,2}$ can easily be found inside of $\mathcal{G}_{n,n}$ by considering two sets $S'_1 \subseteq \binom{S_1}{2}$ and $S'_2 \subseteq \binom{S_2}{2}$ and taking the *restriction* [4] of $\mathcal{G}_{n,n}$ to $S_1 \cup S_2$. Let us also note that an edge between two sets $s_1, s_2 \in \mathcal{G}_{n,n}$ only appears in the induced copy generated by the restriction of $\mathcal{G}_{n,n}$ to $s_1 \cup s_2$ (which is of cardinality 4). Therefore, taking all possible choices of S'_1, S'_2 and the induced copies of $\mathcal{G}_{2,2}$ to which they corresponds results in a complete induced $\mathcal{G}_{2,2} \approx 2K_2$-decomposition of $\mathcal{G}_{n,n}$.

2.2.3 Induced decompositions of dense graphs into small graphs H

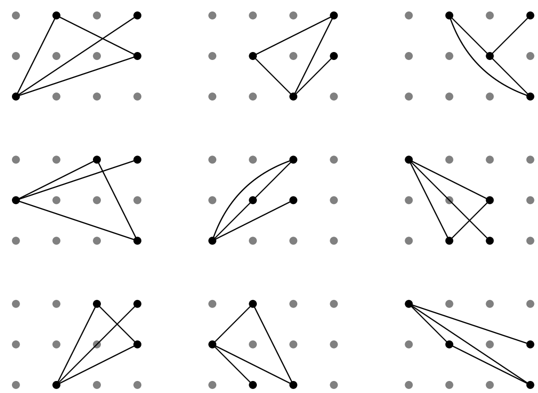
We are now again in front of a dense family of graphs which can be decomposed into $2K_2$. The "next" graph in our family, $\mathcal{G}_{3,2}$ is actually isomorphic to C_6 – which can be decomposed into induced copies of P_4 – and $\mathcal{G}_{4,3}$ can be decomposed into induced copies of $K_{1,3} + e$ as we immediately show.

[3] And if we can hope that they admit one of P_4 or $K_{1,3} + e$ as subgraphs, or even better that they can be decomposed into induced copies of those.

[4] As for the previous construction, by *restriction* we mean considering the graph induced in $\mathcal{G}_{n,n}$ by the sets included in $S_1 \cup S_2$

Decomposing $\mathcal{G}_{4,3}$ into induced copies of $K_{1,3}+e$

The following decomposition was obtained with the software Sage [145], asked to compute a maximum independent set in the graph of all induced $K_{1,3}+e$ subgraphs of $\mathcal{G}_{4,3}$, two of them being adjacent when they share an edge, and thus producing a list of 9 edge-disjoint graphs on 4 edges isomorphic to $K_{1,3}+e$, which partition the 36 edges of $\mathcal{G}_{4,3}$.



Alternatively, if we label the columns with $\{a,b,c,d\}$ and the rows with $\{1,2,3\}$ ($1a$ being the bottom-left corner and $3d$ the top-right one), the decomposition is given by the graphs induced by the following sets of vertices.

$$
\begin{array}{ccc}
\{1a,2d,3b,3d\} & \{1c,2b,3d,2d\} & \{2c,1d,3b,3d\}\\
\{2a,1d,3c,3d\} & \{1a,2b,3c,2c\} & \{3a,1b,2c,1c\}\\
\{1b,2d,3c,3d\} & \{2a,1c,3b,1b\} & \{3a,1d,2b,2d\}
\end{array}
$$

Hence, in order to produce decompositions of dense graphs into copies of P_4, C_6 or $K_{1,3}+e$, it is enough to show that large graphs $\mathcal{G}_{n,n}$ admit induced $\mathcal{G}_{4,3}$- or $\mathcal{G}_{3,2}$-decompositions.

Given two integers k_1,k_2, we can once more find in a large $\mathcal{G}_{n,n}$ an induced copy of $\mathcal{G}_{k_1,k_2}$ by picking two sets $S_1' \subseteq \binom{S_1}{k_1}, S_2' \subseteq \binom{S_2}{k_2}$ and *restricting* $\mathcal{G}_{n,n}$ to $S_1' \cup S_2'$. Unfortunately, when k_1 or k_2 are greater than 3 the same edge of $\mathcal{G}_{n,n}$ could appear in two different induced copies.

To avoid that, we can do better than pick "all the k_1-subsets of S_1 and all the k_2-subsets of S_2". The property used previously to prove that the induced copies of $2K_2$ were disjoint is that two disjoints sets of $\mathcal{G}_{n,n}$ are contained in exactly one choice of sets S_1', S_2'. Hence, we are looking for a collections of k_1-subsets of S_1 (resp. a collection of k_2-subsets of S_2) such that any pair of elements in S_1 (resp. S_2) appear in exactly one set of our collection.

This property is precisely what describes a decomposition of the complete graph K_n into copies of K_{k_1} : a collection of k_1-subsets of $[n]$ such that any pair of elements appears in exactly one element of our collection. Hence, by Wilson's theorem there exists for any integers $k_1,k_2 \geq 2$ an integer α[5] such

[5]One can choose $\alpha = lcm\big(2\binom{k_1}{2}, 2\binom{k_2}{2}, k_1-1, k_2-1\big)$ to trivially fulfill the divisibility constraints from Wilson's theorem, though many other solutions exist in general.. For instance, with $k_1=2, k_2=3$ the decomposition is possible

that for any sufficiently large integer n with $n \equiv 1[\alpha]$ there exist two families $\mathcal{C}_1 \subseteq \binom{S_1}{k_1}, \mathcal{C}_2 \subseteq \binom{S_1}{k_2}$ with the property that any two disjoints sets of $\mathcal{G}_{n,n}$ are contained in the union of two unique elements $S_1 \in C_1$ and $S_2 \in C_2$.

2.2.4 Enlarging our family of dense graphs

The sequence of dense graphs that we have slowly built over the previous decompositions, is, unfortunately, quite unable to lead us further. Indeed, the graph $\mathcal{G}_{n,n}$ remains the "complement of the line graph of a bipartite graph". Being the complement of a line graph, it can not contain as induced subgraphs the complement of the forbidden subgraphs of line graphs (see p.26), and so, for instance, cannot contain $K_1 + K_3$ as an induced subgraph (the complement of the claw graph).

Hence, we can not hope to use this family of graphs to prove that $ex(n, F) = \binom{n}{2} - o(n^2)$ for any graph F. We can, however, try to generalize it so that it may contain any given graph H as an induced subgraph. This can be achieved by considering sets of size 3 or more.

Definition. Let $S^1, \ldots, S^k$ be disjoints sets of cardinality n. The graph $\mathcal{G}_{k\times n}$ is defined over all the sets $S \subseteq S^1 \cup \cdots \cup S^k$ such that $|S \cap S^i| = 1$ for all i. In this graph, two sets $S_1, S_2 \in V(\mathcal{G}_{k\times n})$ are adjacent if and only if $S_1 \cap S_2 = \emptyset$.

When k is fixed, the family of graphs $\mathcal{G}_{k\times n}$ – which depends on n – is asymptotically dense. Indeed, $\mathcal{G}_{k\times n}$ is a $(n-1)^k$-regular graph defined on n^k vertices. Besides, by construction, $\mathcal{G}_{k\times n}$ is trivially vertex-transitive and edge-transitive[6]. The most interesting feature of this graph class is, however, that *any* graph H appears as the induced subgraph of some $\mathcal{G}_{k\times n}$. The complement K_1+K_3 of the claw graph $K_{1,3}$, which did not appear in any of the graphs $\mathcal{G}_{n,n}$ we used in the previous sections, is an induced subgraph of $\mathcal{G}_{3\times 3}$ (see Fig.2.6).

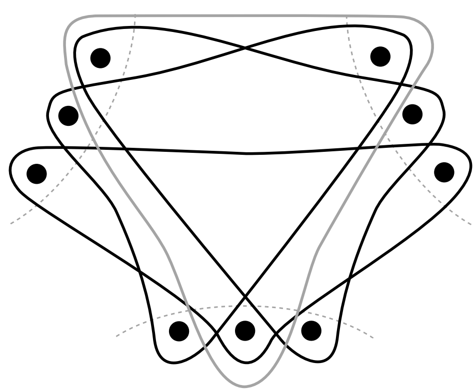


Figure 2.6: A realisation of $K_1 + K_3$ as an induced subgraph of $\mathcal{G}_{3\times 3}$

whenever $n \equiv 1, 3[6]$, and it is possible with $k_1 = 3, k_2 = 4$ whenever $n \equiv 1[12]$. In general, complete characterizations of the values for which there exists a K_k-factorization of K_n are available [54] when $k \leq 9$.

[6]A graph G is said to be *vertex-transitive* (resp. *edge-transitive*) whenever there exists, for any pair $u, v \in V(G)$ of vertices (resp. for any pair $e, e' \in E(G)$ of edges), an automorphism f of G such that $f(u) = v$ (resp. such that $f(e) = e'$). The reader can refer to Godsil [86] for any related question. The papers [118, 95], which explain Brendan McKay's isomorphism algorithm Nauty [117], provide very good excuses to study the basics of Algebraic Graph Theory.

Lemma. *For any graph H there exists integers n_0, k_0 such that H is an induced subgraph of any $\mathcal{G}_{k\times n}$ with $k \geq k_0$ and $n \geq n_0$.*

Proof. It is sufficient (and much more than is actually needed) to choose $n \geq |V(H)|$ and $k \geq \binom{n}{2}$. By definition, there exists in $\mathcal{G}_{k\times n}$ an independent set of size n, i.e. n disjoint sets $H_v \in V(\mathcal{G}_{k\times n})$ of size k representing each vertex $v \in H$. These n sets induce in $\mathcal{G}_{k\times n}$ a complete graph, and they can be modified step by step so as to induce in $\mathcal{G}_{k\times n}$ a copy of H.

Indeed, one can consider in turn each *non*-edge of H, i.e. all of the $\{u_1v_1, \ldots, u_{|E(\overline{H})|}v_{|E(\overline{H})|}\} = E(\overline{H})$, and replace either

$$H_{u_i} \text{ by } (H_{u_i} \cap \overline{S^i}) \cup (H_{v_i} \cap S^i)$$

or

$$H_{v_i} \text{ by } (H_{v_i} \cap \overline{S^i}) \cup (H_{u_i} \cap S^i)$$

so that H_{u_i} and H_{u_i} intersect in S^i. At the end of this procedure, the graph induced in $\mathcal{G}_{k\times n}$ by the sets $(H_v)_{v\in H}$ is precisely H. □

This construction actually requires much larger graphs $\mathcal{G}_{k\times n}$ than are actually needed. Of course, picking $k \geq |E(\overline{H})|$ instead of $k \geq \binom{n}{2}$ would be sufficient, but even there this construction requires for $\overline{K_n}$ a graph as large as $\mathcal{G}_{n\times\binom{n}{2}}$, while $\mathcal{G}_{(n+1)\times n}$ or $\mathcal{G}_{(\lceil log_2(n)\rceil+1)\times 2}$ both contain independent sets of size n.

This being said, we are now in possession of a class of highly symmetrical dense graphs containing any graph H as an induced subgraph.

2.2.5 Induced H-decompositions of dense graphs

Considering the construction of an induced subgraph H in an arbitrarily large graph $\mathcal{G}_{k\times n}$, one can already expect to find many disjoints instances of H inside of it. It is quite unlikely, though, that for a fixed k_H all the sufficiently large graphs $\mathcal{G}_{k_H\times n}$ would admit an induced H-decomposition, if only because of arithmetical constraints. There is no appearant reason why Wilson's theorems would not be required anymore to obtain decompositions of $\mathcal{G}_{k_H\times n}$ when k becomes greater than 3, and so we should either begin to carefully define for which values of n an induced H-decomposition of $\mathcal{G}_{k\times n}$ could exist, or slightly change our proof's methodology.

Indeed, if trying to decompose $\mathcal{G}_{k\times n}$ into induced copies of a graph H may be asking for too much, it is actually sufficient to be able to decompose a *dense subgraph* of $\mathcal{G}_{k\times n}$ into copies of H to obtain our desired result. Hence, we could be satisfied with finding edge-disjoint induced copies of H in $\mathcal{G}_{k\times n}$ using a total of $(1 - o(1))|E(\mathcal{G}_{k\times n})|$ edges. The union of these copies is a perfectly valid example of dense graph admitting an induced H-decomposition.

In order to obtain this decomposition, we will use a lemma from Frankl and Rödl [82] (see also [13, 87]).

Lemma (Frankl and Rödl [82])**.** *Let the integer $m \geq 3$ and the real $z > 3$ be fixed. Then for every fixed $\epsilon > 0$ there exists a real $\delta > 0$ and a threshold value $M_0(\epsilon)$ with the following property. If an m-uniform hypergraph $\mathcal{H}$ has $M > M_0(\epsilon)$ vertices and all of its vertex degrees are between $(1-\delta)d$ and $(1+\delta)d$ for some d, and moreover each vertex pair is contained in at most $d/(logM)^z$ edges of $\mathcal{H}$, then $\mathcal{H}$ contains at least $M/m - \epsilon M$ mutually disjoint edges.*

This wonderful lemma can let us find the desired set of edge-disjoint induced copies of H in a large $\mathcal{G}_{k\times n}$ if we can achieve to rephrase our problem in terms of a maximum hypergraph matching (a matching in an hypergraph is a set of vertex-disjoint edges). What we need now is to define a hypergraph whose edges are all the *occurrences* of H in $\mathcal{G}_{k\times n}$.

If we were to define on $S^1 \cup \cdots \cup S^k$ the hypergraph in which the edges are all the $\bigcup \mathcal{H}$, for any $\mathcal{H} \subseteq V(\mathcal{G}_{k\times n})$ inducing a copy of H in $\mathcal{G}_{k\times n}$, Frankl and Rödl's lemma would not lead us very far. Indeed, a maximum matching is this hypergraph corresponds to a maximum collection of induced *vertex*-disjoints copies of H, i.e. a very sparse graph, while we are looking for edge-disjoint copies of H covering the maximum possible amount of edges in $\mathcal{G}_{k\times n}$.

To do so, we have no other alternative but to define our hypergraph using as a vertex set the *edge* set of $\mathcal{G}_{k\times n}$. We are now working on the hypergraph $\mathcal{G}^H_{k\times n}$, whose set of vertices is $E(\mathcal{G}_{k\times n})$, and whose edges are all the $E(F)$, for any induced subgraph $F \subseteq_i E(\mathcal{G}_{k\times n})$ isomorphic to H.

This hypergraph is vertex-transitive – a direct result of the edge-transitivity of $\mathcal{G}_{k\times n}$ – as well as $|E(H)|$-uniform. Besides, a matching of size $(1-o(1))|E(\mathcal{G}_{k\times n})|/|E(H)|$ corresponds to a collection of induced edge-disjoint copies of H in $\mathcal{G}_{k\times n}$ covering $(1-o(1))|E(\mathcal{G}_{k\times n})|$ edges, i.e. a dense subgraph of $\mathcal{G}_{k\times n}$ admitting an induced H-decomposition.

Hence, a successful application of the previous lemma would yield the equality $ex(n, H) = \binom{n}{2} - o(n^2)$.

Using Frankl and Rödl's lemma

Frankl and Rödl's lemma requires the hypergraph on which it is applied to be "almost regular", i.e. that the vertex degrees be "between $(1-\delta)d$ and $(1+\delta)d$ for some d", which is trivially true in the present case as $\mathcal{G}^H_{k\times n}$ is regular. The lemma then rests on a comparison between the degree of a vertex, and the degree of a *pair* of vertices, i.e. a comparison between the number of edges containing a vertex, and the number of edges containing at the same time the two elements of a pair of vertices.

This second property cannot be obtained "for free" from the vertex-transitivity of the edge-transitivity of $\mathcal{G}_{k\times n}$. What is required here is a relationship between the number of induced copies of H containing a given edge of $\mathcal{G}_{k\times n}$ (i.e. the degree of a given vertex of $\mathcal{G}^H_{k\times n}$), and the number of induced copies of H containing two given edges of $\mathcal{G}_{k\times n}$ (i.e. the number of edges of $\mathcal{G}^H_{k\times n}$ containing two given vertices).

Let us consider two edges $v_1v_1', v_2v_2' \in E(\mathcal{G}_{k\times n})$, corresponding to sets $S_1, S_1', S_2, S_2' \in V(\mathcal{G}_{k\times n})$. By definition, $S_1 \cap S_1' = \emptyset$ and $S_2 \cap S_2' = \emptyset$, but $S_1 \cup S_1'$ and $S_2 \cup S_2'$ may very well intersect.

When $k \geq \binom{|VH|}{2}$, and so when the size of the sets defining the vertices of $\mathcal{G}_{k\times n}$ is large compared to the number of edges of H, much of the information they contain is not relevant to encode "as few as" a $\binom{|VH|}{2}$-size information. In particular, we will be interested in the elements of $S_1 \cup S_1'$ and $S_2 \cup S_2'$ which are *necessary* to encode the information of a given copy of H in $\mathcal{G}_{k\times n}$.

For any non-edge of our copy of H, corresponding to two sets of $\mathcal{G}_{k\times n}$ intersecting on (at least) one element x, let us *mark* x if it belongs to $S_1 \cup S_1' \cup S_2 \cup S_2'$. Having marked at most one element for any non-edge of H, and arguing of the fact that H is not an independent set, there are in each of our four sets at least $k - \binom{|V(H)|}{2} + 1$ unmarked elements.

The unmarked elements of S_1 and S_1' are not actually relevant to the adjacency properties of these sets with the other sets of the copy of H. Hence, we can obtain a different induced copy of H containing S_2, S_2' but avoiding S_1 and S_1' by replacing in S_1 and S_1' these unmarked elements by any other element not contained in the current copy of H, i.e. at least $n - |V(H)|$ alternatives for each of the $2(k - \binom{|VH|}{2} + 1)$ unmarked elements.

As a conclusion, one can associate to any copy of H containing all of S_1, S_1', S_2, S_2' at least $(n - |V(H)|)^{2(k-\binom{|VH|}{2}+1)}$ copies of H not containing S_1, S_1'. As the number of times each of these alternative copies of H can appear from an original copy of k containig all of S_1, S_1', S_2, S_2' is a function of k and H, we deduce that in $\mathcal{G}_{k\times n}$ the degree of a vertex is at least $\Theta(n^{2(k-\binom{|VH|}{2})+1)})$ times the number of edges containing any pair of vertices. Let us only remember that for any $c > 0$ we can chose k_c such that the degree of a vertex is at least $\Theta(n^c)$ times the number of edges containing any pair of vertices.

It is now possible to use Frankl and Rödl's lemma. Indeed, given the family of graphs $\mathcal{G}_{k_c\times n}$ depending on n, the r-regular hypergraph $\mathcal{G}^H_{k_c\times n}$ defined on a vertex set of size $|E(\mathcal{G}_{k_c\times n})|$ verifies that each pair of vertices is contained in at most $\Theta(r/n^c) = \Theta(r/|\mathcal{G}_{k_c\times n}|^{c/2k}) = o(r/log(|\mathcal{G}_{k_c\times n}|)^3)$.

Hence, by this lemma there exists in $\mathcal{G}^H_{k_c\times n}$ a matching covering $(1 - o(n))|E(\mathcal{G}_{k_c\times n})|$ vertices, and in turn there exists in $\mathcal{G}_{k\times n}$ a dense subgraph admitting an induced H-decomposition.

Theorem (Cohen, Tuza [121]). *$ex(n, H) = \binom{n}{2} - o(n^2)$ for any non-empty graph H.*

Chapter 3

Algorithms

In this chapter are presented works whose considerations shift from graph theory to the design of algorithms. In turn are presented an algorithm based on color coding to detect the existence of specific subdigraphs, a scheduling problem in telecommunication networks, and a coloring called *good edge labellings* initially defined as an attack of routing problems in networks.

3.1 Subgraph detection using color coding

In [50] (see Appendix p.142), Fomin, Gutin, Kim, Saurabh, Yeo and myself designed an algorithm whose purpose is to detect the presence of specific subdigraphs. In particular, if we call *out-arborescence* or *out-tree* a digraph obtained from a rooted tree by orienting all its edges away from the root, we obtained the following result.

Theorem (Cohen, Fomin, Gutin, Kim, Saurabh, Yeo [50] (see Appendix p.142))**.** *A copy (understood as a subdigraph) of an out-tree on k vertices can be detected in a digraph on n vertices in $O(n^2 5.704^k)$ time by a randomized algorithm.*

To design this algorithm, we used a general method named *color coding* [14] (see the next section) initially presented as a way to detect treewidth-bounded subgraphs. These algorithms can in turn easily be derandomized at the cost of some efficiency, and yield a deterministic algorithm.

Theorem (Cohen, Fomin, Gutin, Kim, Saurabh, Yeo [50] (see Appendix p.142))**.** *A copy (understood as a subdigraph) of an out-tree on k vertices can be detected in a digraph on n vertices in $O(n^2 6.14^k)$ time by a deterministic algorithm.*

In the following sections are presented the method called *color coding*, an explanation of how they can be derandomized, and a short explanation of *FPT complexity* inside of which our result naturally fits.

3.1.1 Color coding

In 1995, Alon Yuster and Zwick [14] first devised an entertaining technique to locate in a graph G a subgraph isomorphic to a tree T. This method, that they call *color coding*, can be seen as an improvement on the following highly memory-consuming algorithm :

In order to find a tree T in a graph G, pick an edge $uv \in T$, whose removal splits T into two subtrees T_u and T_v. Then, compute for each vertex $r \in G$ the list of all copies of T_u (resp. T_v) in G where r plays the role of u (resp. v). Once done, and for each edge $r_u r_v \in G$, tests whether there exists two disjoint copies of T_u and T_v in which r_u acts as u and r_v as v.

One would naturally expect this algorithm to spend most of its time building, and then exploring the list of pairs of trees T_v and T_u in order to test whether two of them could be disjoint and correctly linked, as the number of such trees grows polynomially according to n. The hang of *color coding* is to use a much smaller (though incomplete) amount of information.

Let the colors appear. Before running the algorithm, all the vertices are given a color among a set of $k = |V(T)|$. The algorithm, instead of using the possibly large information of all the copies of T_u (where r_u acts as u), only remembers "the different sets S of colors such that there exists a copy T_u whose colors belong to S". This is sufficient to measure the complexity of the algorithm using k instead of n. Indeed, if the number of copies can be polynomially large according to n, there is at most 2^k different sets of color they can use [1]. With this information only, it remains possible to identify pairs of vertex-disjoint trees T_u and T_v : if we know there is a copy of T_u "around" r_u and a copy T_v "around" r_v such that the colors used in T_u are disjoint from the colors used in T_v, then obviously the trees themselves are disjoint.

In this happy context, the algorithm can return an answer. Most of the other times, or for example when all the vertices are given the same color, this algorithm will miss most – if not all – pairs of valid trees. Sheer optimism not being sufficient when it comes to finding a tree in a graph, we now have to pay the dependency on n which disappeared when the list of all copies of subgraphs has been shortened to a list of different color sets.

In [14], the authors find their way out by exploring the space of different colorings and note that such a modified version finds a copy of T in G on the condition that all of its vertices are initially given different colors, which happens with probability $k!/k^k$ when the vertices of G are colored uniformly at random. Hence, given an integer k, this algorithm finds in any graph G a copy of a tree T on k vertices with probability at most $k!/k^k$, or equivalently needs to be run at most $1/log(1 - \frac{k^k}{k!})$ times to find a subgraph T with probability at least $1/2$.

Derandomization – covering arrays

From this random algorithm, Alon Yuster and Zwick [14] then extract a deterministic one. This can be done in many ways, the first one being to sequentially consider all the possible n^k colorings of $V(G)$, which would result in a very poor algorithm (it is far less expensive to test whether any of the $\binom{n}{k}$ subsets of vertices represents our tree!). As the algorithm applied on a given coloring only requires the vertices of the copy of T to be given different colors in order to detect it, one could first attempt to reduce the space of different colorings through the use of covering arrays (see [96] for a general survey and [106] for a survey focused on binary covering arrays).

Definition (Covering Arrays)**.** A covering array of strength t from $\{1, \dots, n\}$ to $\{1, \dots, k\}$ of size s is (equivalently) :

- An $s \times n$ matrix M with values in $\{1, \dots, k\}$ such that the restriction of M to any t columns contains rows corresponding to all the possible k^t vectors of length t on $\{1, \dots, k\}$.

[1] Actually $\binom{n}{|T_u|}$

- A collection $\mathcal{C}$ of s functions from $\{1,\ldots,n\}$ to $\{1,\ldots,k\}$ such that any assignment of values in $\{1,\ldots,k\}$ to the elements of a t-set of $\{1,\ldots,n\}$ is realized by at least one function of $\mathcal{C}$.

Hence, considering as a collection of colorings a (smallest possible) covering array of strength k from $\{1,\ldots,n\}$ to $\{1,\ldots,k\}$ would certify that any copy of T in G would be detected by the *color-coding* algorithm, as any k-subset would be assigned all of its possible k-coloring, including the $k!$ colorings in which they are all assigned different colors.

In their application, the authors instead use [14] families of *perfect hash functions* [8], which are precisely what their algorithm requires.

Definition (Families of hash functions)**.** A function $f : [m] \mapsto [n]$ is said to be a S-perfect hash function for a set $S \subseteq [m]$ if $|f(S)| = |S|$. A family of hashing functions $\{f_1,\ldots,f_s\}$ is k-perfect if it contains a S-perfect hash function for any set S of size k.

With such a set of colorings, there exists for every k-subset S of $[n]$ a function f_i such that all the elements are assigned different colors). They then explain how to build a family of k-perfect hash functions from $\{1,\ldots,n\}$ to $\{1,\ldots,k\}$ of size $2^{O(k)}log(n)$ by composing a construction of k-perfect hashing functions from $\{1,\ldots,n\}$ to $\{1,\ldots,k^2\}$ [141] and k-perfect hashing functions from $\{1,\ldots,k^2\}$ to $\{1,\ldots,k\}$, obtained by Naor et al. [123] and Alon et al. [6].

As in our situation [50] (see Appendix p.142) the algorithm is built recursively, we instead make repeated use of binary covering arrays.

3.1.2 FPT Complexity

The previous algorithm can be seen as an attempt to study subgraph detection in the framework of *FPT complexity*. Indeed, by improving upon the algorithm presented on p.64, Alon Yuster and Zwick *transferred* this algorithm's complexity from the size of G to the size of the subgraph H they want to find : if it was originally of complexity $O(|G|^{\Theta(|H|)})$ (one can sometimes find $\Theta(n^k)$ copies of a graph on k vertices in a graph of n vertices), the color coding method reduced it to a runtime of the form $f(|H|)|G|^c$ – where c is a constant – so that the algorithm's asymptotic complexity does not depend anymore on the value of the fixed graph F.

The existence of such an algorithm could lead to think that the complexity of finding a copy of H in G lies more in the size of H than in the size of G, as f is exponential while $|G|^c$ is a fixed polynomial. For this reason the analysis of an algorithm through the lens of FPT complexity can be seen as an attempt to get a better theoretical understanding of the source of computational hardness in a given problem, and provides a framework refining classical complexity when all it yields is NP-Hardness.

More practically, let us look at the Maximum Independent Set problem, which amounts to decide – given a graph – the size of a maximum set of pairwise nonadjacent vertices. This problem can trivially be solved by testing all of its $2^{|V(G)|}$ subsets and keeping in memory the size of the maximum independent set found, which takes time at most $2^{|V(G)|}|E(G)|$. As a decision problem, it is usually formulated as:

INPUT : A graph G, an integer k
OUTPUT : Does there exist in G an independent set of size at least k ?

There is a algorithm solving this problem in time $\binom{|V(G)|}{k}|E(G)| \leq |E|n^k$, exactly as previously – by testing all the k-subsets of n, which teaches us nothing new. We barely used the information given by k to produce this different algorithm, and the point of FPT complexity is precisely to define a new analytic framework to decide which kind of parameters give us a powerful information on how to solve the problem.

A parameterized computational problem is said to be FPT (Fixed Parameter Tractable) if its running time is of the form $O(f(k)n^c)$ for some constant c. This can be read as a sign that the problem's complexity lies more in what the parameter k measures[2] than in the actual size of the instance. Under assumptions in FPT complexity which mirror the $P \neq NP$ conjecture, there is actually no FPT-time algorithm solving this parameterized complexity problem[3]. A different choice of parameter may have led to a different conclusion

INPUT : A graph G with tree-width at most k
OUTPUT : Does there exist in G an independent set of size at least t ?

In this different setting, the maximum independent set problem can be solved in $2^{O(k)}|E(G)|$ time [20, 80, 29], which is the complexity of a FPT problem.

3.2 Reconfiguration problem

The WDM reconfiguration problem is a macroscopic[4] scheduling problem appearing in telecommunication networks. This problem appears when different users need to communicate with each other using edge-disjoint routes in a network. Practically, we consider in this setting that several users in a network need to communicate with each other (each user has a list of correspondents with whom he wants to exchange information), and that the paths in the network they need to exchange information are to be edge-disjoint, which is the general definition of the Integer Multiflow problem.

This problem is typically hard to solve, even on instances of moderate size[5], and is in what concerns us part of the instance of the problem. In the WDM reconfiguration problem, we are actually given a network, and two different solutions to the integer multiflow problem of identical parameters (in particular the same pairs of nodes exchange information), and we will be striving to transform this first routing into the second one according to basic rules meant to preserve the quality of service.

In practice, this scheduling problem can be motivated by the need of a maintenance operation on some equipment of the network, during which some part of the network itself may be made unavailable. In order to prevent it, we would ideally like to redistribute the load withstood by the equipment to be interrupted, in such a way that the users observe the least possible perturbation in their exchanges.

The WDM reconfiguration problem is a scheduling problem, whose rules follow. They define two operations modifying a routing scheme, from which precise optimization problems will be defined afterwords.

[2]The FPT complexity of the same optimization problem can be analyzed according to different parameters. Actually, a FPT complexity problem is not properly defined if the parameter is not.

[3]The Independent Set problem, parametrized with k – the size of the desired independent set – belongs to complexity class $W[1]$.

[4]We call it "macroscopic" as the communications we are dealing with are actually aggregated flows of many communications, which will be operated on as monolithic units.

[5]It is also used as a subroutine to detect minors in graphs, which can give an idea of its computational hardness.

WDM reconfiguration problem : the rules

An instance of the WDM reconfiguration problem is a graph, along with two sets S_1, S_2 of edge-disjoint paths corresponding to feasible solutions of the integer multiflow problem. In particular, there is a bijection $f : S_1 \mapsto S_2$ associating to each path of S_1 a path $f(P) \in S_2$ with the same endpoints.

In order to transform a routing scheme S_1 into a routing scheme S_2, the following two operations can be applied on the network, to iteratively modify the "current" routing.

- Interrupt a connection, which frees the resources (=edges) it uses

  (in this case any exchange of information through connection are to be interrupted)

- Replace a path $P \in S_1$ (currently in use, or interrupted) by $f(P)$ if the resources needed by $f(P)$ are available

  (in which case its endpoints can immediately use it to exchange information)

Dependency digraph – second formulation

Given two solutions to the integer multiflow problem, the only relevant information – according to the rules of the WDM reconfiguration problem – lies in a (non-symmetric) binary relation. Two paths $P_1, P_2 \in S_1$ are in conflict whenever P_1 is to be re-routed on a path $f(P_1) \in S_2$ which requires resources used by P_2.

From an instance of the WDM reconfiguration problem, one can then build a dependency digraph, in which the vertices represent the paths of S_1, with an edge from a vertex u to a vertex v whenever u is in conflict with v. In this new graph, the WDM reconfiguration problem can be re-defined as a *cops and robbers* game [128].

WDM reconfiguration problem : in the dependency digraph

Inside of the dependency digraph, a reconfiguration scheduling is equivalent to a sequence of the following moves :

- Add an agent on a vertex (i.e. interrupt the corresponding connection)

- Remove a vertex – and eventually the agent lying on top of it – if all of its outneighbors (possibly 0) have agents placed on them.

This game being completely equivalent to the previous reconfiguration rules, it is equivalent to study the dependency digraph instead of sets of edge-disjoint paths.

One can be led to wonder whether the graphs obtained by this transformation may have any useful specific properties that could be used of help in the analysis. Fortunately, any digraph can be obtained as the dependency digraph of a pair of two equivalent routings [52] and so the two formulations totally equivalent.

Finding good reconfiguration orderings

Having modelled the routing reconfiguration problem with a digraph, the next step is to formalize the objectives of a good reconfiguration ordering and make it an optimization problem. Indeed, there is always a trivial solution to a reconfiguration problem : it is enough to interrupt all the communications at once, which frees all the resources, and to resume the information exchange by using the path $f(P)$ when $P \in S_1$ was formerly used. It would have been much better to be able to set up the path $f(P)$ before P is interrupted, as in this case the communications can be *moved* from P to $f(P)$ unbeknownst to the users.

For this reason, it makes sense to define and study various metrics of the impact a scheduling may have on the quality of service – for instance the total number of communications interrupted or (to a lesser extend) the maximum number of simultaneously interrupted communications. With Coudert, Mazauric, Nepomuceno and Nisse [52] (see Appendix p.161), we studied the computational complexity of determining optimal orderings minimizing those metrics, as well as their tradeoffs.

Minimizing the number of agents simultaneously present in the digraph (which amounts to minimizing the number of simultaneously interrupted connections in the network) has many common points with determining the pathwidth of a graph [56]. This similarity appears when computing a reconfiguration ordering of a digraph obtained from a graph after replacing each edge by two opposed arcs. By minimizing the total number of agents appearing in the digraph (i.e. by trying to minimize the total number of connections that will be interrupted), we actually solve the Feedback Arc Set problem[6] as no digraph containing a circuit C can be processed if no agent ever touches a vertex of C. From these two metrics, we defined constrained problems in which one is to minimize the number of agents simultaneously present, at the same time using a bounded number of agents in total. Similarly, given a constraint of the maximum number of agents simultaneously present, one can strive to minimize the total number of agents used by a reconfiguration ordering. Unfortunately for practical applications, these problems often turn out to be NP-Hard, as well as hard to approximate.

3.3 Good edge Labelling

The problem presented in this third section also originates from a routing problem.

In 2009, Bermond, Cosnard and Pérennes [28] studied the problem of allocating frequencies to lightpaths in fiber networks. As in the previous section, the main problem in this context remains to find in a network the actual routing of communications between pairs of nodes willing to exchange information with each other, but once this routing is found many other NP-Hard problems are still left to be solved. In the context of fiber networks, several communications can share the same medium on the condition that they use different frequencies. While this constraint is *local* (all the communications using the same edge use different frequencies) is it expensive (both in equipment and delay) to alter a lightpath so as to change its frequency.

[6]The Minimum Feedback Arc Set problem is a covering problem : the goal is to find in a digraph a smallest set of arcs hitting all the circuits.

For this reason, Bermond et al. considered this problem with the additional constraint that a given communication has a constant frequency. In this context, everything boils down to a coloring problem. Indeed, given a routing of the communications (some of which sharing edges, according to their capacity), the assignment of frequencies such that any two lightpaths sharing a medium use different frequencies is the very definition of a proper coloring in a *conflict graph*. This is, in practice, easier to solve than the routing problem. From there, Bermond et al. wondered in which situations the coloring problem could be the hardest of the two : it is trivially harder in a class of directed graphs in which there exists only one path from one vertex to another. The routing problem is then trivial, but on such constrained graphs the resulting coloring problem could also have lost its complexity.

Hence, they began the study of the conflict graph produced by a (trivial) routing in a digraph having this uniqueness property. They found out [28] that this conflict graph admitted what they called a *good edge labelling* (see below), and that conversely any graph which admitted a *good edge labelling* could be produced as the conflict graph of a communication routing in a constrained digraph. In their definition of a *good edge labelling* of a conflict graph, the idea of unique paths remains.

Definition. A *good edge labelling* of a graph is an assignation of numerical values to its edges such that between any two vertices u and v there exists at most one increasing path from u to v in which the edge labels are increasing.

The class of graphs admitting such an edge labelling – and so the class of conflict graphs produced by routings in their constrained digraphs – has several basic properties which may be helpful in solving the complementary proper coloring problem. A triangle, for instance, does not admit a good edge labelling, and so graphs admitting one are necessarily triangle-free. Besides, as there is in any edge labelling of a P_3 an increasing path from one endpoint to the other, any graph containing two vertices linked by three different P_3 (i.e. a graph containing a copy of $K_{2,3}$) does not admit a good edge labelling.

In [19] (see Appendix p.185), we studied some properties of the graphs admitting a good edge labelling and their recognition :

Theorem (Araujo, Cohen, Giroire, Havet [19])**.** *Deciding whether a bipartite graph admits a good edge labelling is NP-Complete.*

As a corollary of this result, and under the assumption that $P \neq NP$, the class of graphs admitting a good edge labelling can not only consist in $K_3, K_{2,3}$-free graphs, as recognising those can be done in time $O(n^5)$ using an enumerative algorithm. Indeed, we also presented in [19] an infinite family of incomparable $K_3, K_{2,3}$-free graphs admitting no good edge labelling (see Fig.3.1).

Besides, with the remark that a minimal graph admitting no good edge labelling can not contain a *matching cut* (i.e. a set of edges disconnecting the graph and inducing a matching), we also proved that some classes of graphs always admit a good edge labelling, i.e. triangle-free outerplanar graphs, planar graphs of girth at least 6, subcubic $K_3, K_{2,3}$-free graphs, and $K_3, K_{2,3}$-free ABC-graphs (a class of graphs with no matching cut defined by Farley and Proskurowski [76]).

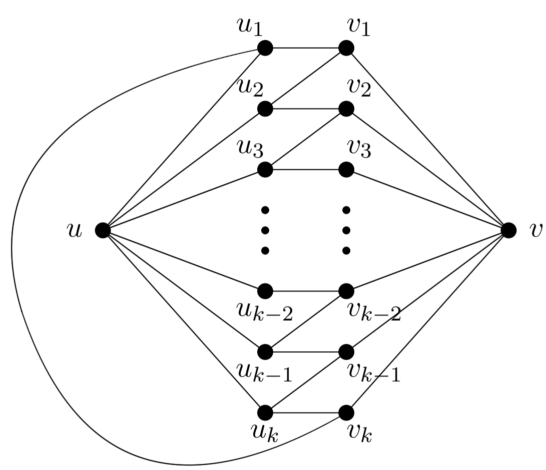


Figure 3.1: Family of incomparable graphs not admitting any good edge labelling

Chapter 4

Wiener and Zagreb indices in chemical graphs

The Wiener and Zagreb indices are not two, but three indices defined over graphs, measuring properties of interest in chemistry. Unusually enough, they are not NP-Complete to compute, and can be easily evaluated on any graph, depending only on very basic parameters. The Wiener index is actually the sum of the distances between all pairs of vertices of a graph – for obvious reasons defined on connected graphs.

$$W(G) = \sum_{u,v \in V(G)} d(u,v)$$

In chemistry, this measure belongs to the larger class of the so-called *topological indices* [131]. Researchers also studied the behavior of the Wiener index of line graphs (called *edge-Wiener index* in [102]), and then compared the respective values of $W(G)$ and $W(L(G))$ on a given graph G ([60, 103, 65, 63, 66]).

The Zagreb indices were introduced in 1972 by Gutman et al. [90], and have been used since as a tool to study various molecular parameters. As graph measures, they are defined by

$$M_1(G) = \sum_{v \in V(G)} d(v)^2 \quad \text{and} \quad M_2(G) = \sum_{uv \in E(G)} d(u)d(v)$$

Since [126], there has been a growing interest in comparing the parameters $\frac{M_1(G)}{n}$ and $\frac{M_2(G)}{m}$ on various classes of graphs.

Wiener index

In [49] (see Appendix p.219), we proved that a minimalistic assumption on the degrees of a graph is sufficient to ensure $W(G) \leq W(L(G))$.

Theorem (Cohen, Dimitrov, Krakovski, Škrekovski, Vukašinović)**.** *If the minimum degree of a graph G is at least two, then $W(G) \leq W(L(G))$.*

This proof is essentially analytic, and derived from an inequality which lets us compute two measures depending exclusively on the distance between vertices, instead of working on two different graphs or dealing with distances between both edges and vertices. If we denote $e = uv, e' = u'v' \in E(G)$, then

$$d_{L(G)}(e, e') \geq \frac{1}{4}\Big[d(u, u') + d(u, v') + d(v, u') + d(v, v')\Big].$$

From this, one can then derive the inequality $W(G) \leq W(L(G))$ by assuming a minimum degree of at least 2 and some appropriate counting.

While, cycles are the only graphs of minimum degree 2 for which the inequality holds, many instances other instances exist which satisfy $W(G) = W(L(G))$ – and those have leaves. In particular, Dobrynin and Mel'nikov posed the following problem.

Problem (Dobrynin and Mel'nikov[64])**.** Is it true that for every integer $g \geq 5$, there exists a graph $G \neq C_g$ of girth g, for which $W(G) = W(L(G))$?

We partially answered this question with the following result [49].

Theorem (Cohen, Dimitrov, Krakovski, Skrekovski, Vukasinovic)**.** *There exists a graph G of arbitrarily large girth satisfying $W(G) = W(L(G))$.*

The construction of these graphs is pretty elementary, as they are built from a cycle (with a chord) to which are attached two paths of variable lengths (see Fig.4.1).

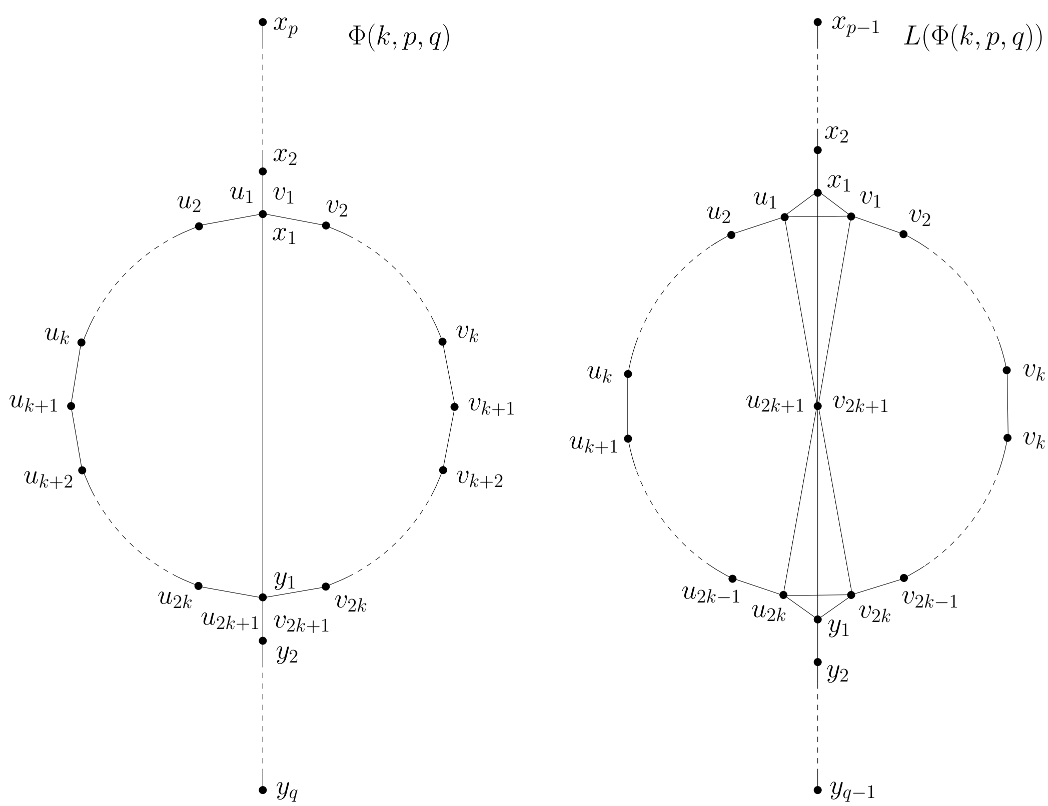


Figure 4.1: Graphs $\Phi(k, p, q)$ and $L(\Phi(k, p, q))$

k	p	q	$W(G)$	$W(L(G))$
2	2	2	96	106
2	2	3	128	136
2	2	4	171	178
2	2	5	226	233
2	3	2	128	136
2	3	3	165	170
2	3	4	214	217
2	3	5	276	278
2	4	2	171	178
2	4	3	214	217
2	4	4	270	270
2	4	5	340	338
2	5	2	226	233
2	5	3	276	278
2	5	4	340	338
2	5	5	419	414
3	2	2	254	274
3	2	3	306	322
3	2	4	373	386
3	2	5	456	467
3	3	2	306	322
3	3	3	363	374
3	3	4	436	443
3	3	5	526	530
3	4	2	373	386
3	4	3	436	443
3	4	4	516	518
3	4	5	614	612
3	5	2	456	467
3	5	3	526	530
3	5	4	614	612
3	5	5	721	714

k	p	q	$W(G)$	$W(L(G))$
4	2	2	534	568
4	2	3	610	638
4	2	4	705	728
4	2	5	820	839
4	3	2	610	638
4	3	3	691	712
4	3	4	792	807
4	3	5	914	924
4	4	2	705	728
4	4	3	792	807
4	4	4	900	908
4	4	5	1030	1032
4	5	2	820	839
4	5	3	914	924
4	5	4	1030	1032
4	5	5	1169	1164
5	2	2	972	1024
5	2	3	1076	1120
5	2	4	1203	1240
5	2	5	1354	1385
5	3	2	1076	1120
5	3	3	1185	1220
5	3	4	1318	1345
5	3	5	1476	1496
5	4	2	1203	1240
5	4	3	1318	1345
5	4	4	1458	1476
5	4	5	1624	1634
5	5	2	1354	1385
5	5	3	1476	1496
5	5	4	1624	1634
5	5	5	1799	1800

The polynomials obtained with these values are

$$W(\Phi(k,p,q)) =$$

$$6k^3+2k^2p+2k^2q+2kp^2+2kq^2+\frac{1}{2}p^2q+\frac{1}{2}pq^2-k^2+\frac{1}{6}p^3-p^2+\frac{1}{6}q^3-q^2-7k-\frac{7}{6}p-\frac{7}{6}q+4$$

$$W(L(\Phi(k,p,q))) =$$

$$6k^3+2k^2p+2k^2q+2kp^2+2kq^2+\frac{1}{2}p^2q+\frac{1}{2}pq^2+k^2+\frac{1}{6}p^3+\frac{1}{6}q^3-2kp-2kq-\frac{1}{2}p^2-pq-\frac{1}{2}q^2+k+\frac{1}{3}p+\frac{1}{3}q$$

Figure 4.2: Values of $W(\Phi(k,p,q))$ and $W(L(\Phi(k,p,q)))$ for $k,p,q \in \{2,3,4,5\}$

The main part of our work actually consists in computing the values of $W(G)$ and $W(L(G))$ when the three different parameters of our graphs are fixed (each pending path has its own length, and the third parameter measures the size of the central cycle).

While such computations are long and heavy, there exists a quick way to convince oneself that $W(G) = W(L(G))$ holds for many graphs of this class. To this aim, one but needs to believe for a while that – considering the definition of the Wiener index – the value of both $W(G)$ and $W(L(G))$ is a polynomial function depending on our three parameters. If this is true, then of course the difference $W(G) - W(L(G))$ is also a polynomial[1]. Besides, given the small size of our graphs, the parameters $W(G)$ or $W(L(G))$ can be easily evaluated on many instances (see p.73).

Based on these values, and through the additional assumption that the polynomial corresponding to $W(G) - W(L(G))$ is of degree at most 3, one can make great use of Hilbert's Nullstellensatz (see p.23). Indeed, under all those assumptions there exists only one polynomial of our three variables whose value is precisely the one computed directly on 4^3 different graphs of our family (see p.73). Once this polynomial has been computed – through Lagrange's interpolation polynomials, or more easily by exact matrix inversion – it can be easily tested on instances of $\Phi(k, p, q)$ different from those used to define it, to check that it indeed predicts the actual values of $W(G)$ and $W(L(G))$.

Through these successive manipulations, one can derive the existence of graphs satisfying $W(G) = W(L(G))$ by showing the existence in the polynomial $W(G) - W(L(G))$ of infinite families of zeroes.

Zagreb indices

In [17] (see Appendix p.206) and [16] (see Appendix p.230), together with Vesna Andova and Riste Škrekovski, we studied the inequality $M_1/n \leq M_2/m$.

As it trivially holds on regular graphs, we gave in [17] a condition on the span of the degree sequence of a graph ensuring that it verifies this relation.

Theorem (Andova, Cohen, Skrekovski)**.** *If G is a graph with $\Delta(G) - \delta(G) \leq \lceil \sqrt{c}\, \rceil$ and $\delta(G) \geq c$ for some integer c, then $\frac{M_1(G)}{n} \leq \frac{M_2(G)}{m}$.*

We also provided a class on which the opposite holds regardless of the value of Δ, for an infinite sequence of graphs of any maximum degree Δ.

Finally, we gave a short elementary proof of the known inequality $M_1/n \leq M_2/m$ for trees [155] and unicyclic graphs [45].

[1] Note that this assertion can actually hide a mistake. Even though we may intuitively feel that the value of $W(G)$ or $W(L(G))$ is a polynomial function of k, p, q, this polynomial may involves operands like $\lfloor \frac{k}{2} \rfloor$. This kind of integer divisions is by no means a polynomial, and so it may be necessary to study the value of $W(G)$ and $W(L(G))$ under some additional constraints of the congruence classes of the parameters.

Chapter 5

Applied theoretical research – Sage

Like a mechanical hand helping researchers, the assistance provided by computers can be invaluable when working on theoretical problems. Since the beginning of my studies, I have been needing to compute properties – at first mainly proper colorings – of graphs for which computers are able to enumerate all the possibilities. I began by defining my own graph libraries, first in C and in Perl, to finally download and use various more complete libraries from the internet (including the MASCOPT[112] library developed in the MASCOTTE research team). Each one had a different aim : some were good at handling large graphs but were missing important time-consuming methods, other were centered on statistical properties or linear algebra, other only studied graphs for as long as they were computer networks and focused on routing properties, etc... Of course, this meant they all used a different class structure and different languages.

This is how I began to use the general-purpose software Sage [145], which describes itself as *"an attempt to build an open-source alternative to Magma, Maple, Mathematica and Matlab"*, and is a pure product of academia. Being mainly developed by researchers or students (more than 230 contributors), Sage addresses a very wide range of mathematics and grounds many of its algorithms on third-party open-source code developed independently. One of its most appealing features is of being a common front-end to state-of-the-art libraries which are meant to execute one very specific task[1], and to do it accurately and efficiently.

Graph Theory in Sage

Sage has now become a complete and expressive library for the study of graph theory (among many others fields of mathematics). It is able to generate close to 100 different graphs and digraphs[2] on which it can use more than 260 different functions ranging from automorphism test, flows, connectivity and matchings to exact algorithms for NP-hard problems, along with many tests of recognition of different graph classes.

For the purposes of research, I enjoyed writing Sage code related to my studies, and ended up making many additions among the list of graphs it is able to generate, as well as among its graph and digraph methods.

[1]Efficiency often excludes "user-friendliness" – each independent software having its specific inputs/outputs – which is another point for the existence of Sage.

[2]Many of them are "named" graphs, like Petersen's or Chvatal's, but it also contains methods to generate different types of random graphs (G_p^n, G_m^n, random regular graphs, ...) or families of graphs (Kneser's, unlabeled graphs, from a specified degree sequence, ...).

Bipartite graphs from degree sequences	Circuit Graph
Complete Multipartite Graphs	DeBruijn
DirectedGNM	Kneser's Graphs
Random Bipartite Graphs	Random interval graph
The World's graph	

Figure 5.1: Additions to Sage's graph classes

Acyclicity test (with certificates)	Bounded outdegree orientation
Chordality test (with certificates)	Computing graph k-cores
Connectivity methods [3]	Counting spanning trees
Eulerian orientation	Exhaustive search of subgraphs
Gallai tree recognition	Gomory-Hu decomposition
Interval graph recognition	Lex-BFS
Line graph recognition	Max flow
Merging vertices	Minimum out-degree orientation
Overfull graph recognition	Perfect graph recognition[4]
Petersen 2-factor theorem	Split graphs recognition
Shortest path between all pairs[5]	Splitting edges
Strong orientation of graphs	Triangle-free graphs recognition
Szeged index	Wiener index

Figure 5.2: Additions to Sage's graph/digraph methods (LP excluded)

Cython : interfacing Sage and other components

Sage's original language is Python [150], i.e. a highly expressive language in which many scripts can be written through only a few words, though admittedly not the fastest available. Through the use of Cython [144], it is possible to write in Sage code based upon both Python and C/C++, possibly mixed in the same file. Sage's graph structure – originally based upon the graph library NetworkX [125] – has since been reimplemented at near-C levels by Robert Miller with huge gains in efficiency. I also participated to this effort by rewriting at the same level important routines like DFS and BFS traversals as well as classical and bidirectional versions of Dijkstra's algorithm, and the implementation now used by Sage to compute the distance between all pairs of vertices or their associated shortest paths is also used by the graph library Grph [100]. Sage's graph library is today written indifferently in Python, Cython, or C – the language being picked according to the needs in efficiency.

Cython, however, is especially useful to link with Sage pieces of code written independently, in any language able to produce library files (C/C++/Fortran/...). One of my first contributions was a patch exposing several algorithms related to cliques in graphs written in the software Cliquer [127], which now lets Sage quickly compute maximum cliques (equivalently independent sets and vertex

[3] Computation of min vertex/edge cuts, or the paths promised by Menger's theorem, ...

[4] Through search of odd holes and odd antiholes

[5] Many implementations have been made of algorithms solving this problem (Floyd-Warshall, BFS, ...) in order to make it efficient, as developers of the library Grph[100] were at the same time striving to find one able to handle graphs on thousands of nodes which require a large amount of memory

covers). In the same way, Sage can now use Fabien De Montgolfier's implementation [61] of the modular decomposition algorithm he designed with Capelle and Habib [44].

Of even more general use in Graph Theory, my most useful contribution to Sage was probably the implementation of an interface with the LP and MILP solvers GLPK[85] (GPL), Coin Branch-and-Cut[47] (Eclipse Public License), or CPLEX[57] (proprietary) which are now used to solve most of the NP-Hard problems Sage can handle. This interface is also written a C-level.

Linear Programming in Sage

Linear Programming often yields impressive results when exhaustive enumeration would have been unrealistic. With a transparent interface between Sage and LP solver, I was able to implement in Sage many new methods solving NP-Hard optimization or existence problems.

Acyclic edge coloring	Degree-constrained subgraph
Disjoint routed paths	Dominating set
Edge-disjoint spanning trees	Fractional chromatic index
Grundy coloring	H-minor
Independent Set of Representatives	Knapsack
Linear Arboricity	Longest path
Matching	Max cut
Max Flow	Maximum average degree
Min Feedback arc set	Min Feedback vertex
Multiflow	Multiway cut
Steiner Trees	Topological minors
TSP	Vertex/Edge proper coloring

Figure 5.3: Additions to Sage's graph/digraph : Methods based on Linear Programs

With this interface, solving a Maximum Independent Set problem on a graph can be done with the following lines, which is infinitely faster than what would be needed to implement a purely combinatorial algorithm.

```
sage: g = graphs.PetersenGraph()
sage: p = MixedIntegerLinearProgram()
sage: is_used = p.new_variable(binary = True)

sage: p.set_objective( sum(is_used[v] for v in g) )
sage: for u,v,_ in g.edges():
...       p.add_constraint( is_used[u] + is_used[v] <= 1)

sage: p.solve()
```

More complicated Linear Programs involving generation of constraints (useful for the TSP, Fractional Chromatic Index, Minimum feedback arc set, . . .) can also be easily written in a few lines through the same interface, and yielded impressive gains of efficiency, in particular when comparing for the Minimum Feedback Arc set problem the classical *ordering-based* LP formulation and the *covering* formulation, in which one tries to compute a minimum set of edges covering all the cycles in the graph.

Documentation

Sage being an open-source software and developped by a large number of people, it necessarily contains an extensive documentation, both of the code and of the mathematical objects it implements. This documentation is usually more important (in number of lines) than the code itself, and is also used to automatically test the whole Sage library for possible inconsistencies.

With several colleagues, I participated to the writing of a Sage manual [46] for french undergraduates students. This book, published under the Creative Commons license and authored by Alexandre Casamayou, Guillaume Connan, Thierry Dumont, Laurent Fousse, François Maltey, Matthias Meulien, Marc Mezzarobba, Clément Pernet, Nicolas M. Thiéry, Paul Zimmerman and myself, attempts at covering the mathematics those students could meet through the use of Sage, with the hope that students could find there yet other reasons to enjoy mathematics, and possibly eventually participate to the development of the software itself.

5.1 Several new LP formulation of optimization problems

In this chapter are presented several Mixed Integer Linear Programs solving various graph problems having in common constraints of acyclicity or connectedness. They are based on the LP formulation of the maximum average degree (see Sec.0.5.1), which lets us test (from the inside of a linear program) whether a set of edges is a forest, or if it contains a spanning tree, hence testing its connectedness.

This trick is sufficient to write MILP formulations for problems like the Traveling Salesman Problem achieving better performances than through the use of an ordering on the vertices which usually produces bad relaxations. It can also be used to write an exact solver for the H-minor problem even though it remains computationally exhausting. Their second attractive feature is to use a reduced number of constraints : the Traveling Salesman Problem can be solved through MILP with constraint generation, though the corresponding formulations are theoretically of exponential size.

The general principle behind these formulations is conveyed by the following MILP, which computes a minimum-cost spanning tree based on the LP computation the maximum average degree of a graph presented in p.34.

This problem is very easy to solve, as well as easy to implement. The following MILP is admittedly the worst way to solve it, but it conveys the idea that will be used to solve several different hard optimization problems later on.

A subgraph H of a graph G – understood as a set of edges – is a spanning tree of G if and only if it satisfies both $|E(H)| = |V(G)| - 1$ and $mad(H) < 2$. As LP solvers are numerical algorithms subject to noise, there is little meaning is writing a strict constraint "$<$" in a LP, and it has to be replaced by the equivalent version $mad(H) \leq 2 - \frac{2}{n}$. From these two constraints we can define the desired LP.

Minimum spanning tree

- Minimize :

$$\sum_{e \in E(G)} w_e b_e$$

- Such that :
  - The number of edges in a spanning tree is precisely $|V(G)| - 1$:

$$\sum_{e \in E(G)} b_e = |V(G)| - 1$$

  ◇ The set of edges is acyclic :
    * $mad(H)$ is less than $2 - \frac{2}{n}$

$$\forall v \in V(G)$$

$$\sum_{\substack{e \in E(G) \\ e \sim v}} x_{e,v} \leq 2 - \frac{2}{n}$$

    * Each edge sends a flow of 2 to its endpoints if it is in the spanning tree, 0 otherwise :

$$\forall e = uv \in E(G)$$

$$x_{e,u} + x_{e,v} = 2b_e$$

- Variables :
  - b_e is a boolean variable indicating whether the corresponding edge is in the spanning tree
  - $x_{e,v}$ is a real positive variable, representing the flow sent by an edge e to one of its endpoints v.

This use of the LP solving the MAD problem is the key to solving several optimization problems based on connectivity whose list follows. Each of these entries point towards a MILP formulation that has been implemented in the software Sage [145], where they can now be used to solve actual instances.

Some of them, like TSP problem, can be solved in Sage through different means or formulations whose efficiency depends on the instances.

In the given MILP formulations, the constraint $\leq 2 - \frac{2}{n}$ *differs only in the fact that edges send a* *flow of 1 to their endpoints instead of 2. This constraint hence becomes $\leq 1 - \frac{1}{n}$. The terminology "an edge sends a flow of 1 to its endpoints" is kept all along, and appears only at parts of the formulations whose aim is to bound the MAD, or equivalently to ensure that a given set of edges is acyclic.*

Longest path

The Longest Path problem amounts to find in a graph G a path subgraph of longest size.

> **INPUT :** A graph G
> **OUTPUT :** A path $P \subseteq G$ of maximum size.

It can be defined as a Linear Program as a maximum set of edges P such that P is an acyclic subgraph of maximum degree 2 with $|V(P)|$ vertices and $|V(P)| - 1$ edges. The formulation is given on page 84.

Traveling Salesman Problem

The Traveling Salesman Problem is the optimization problem generalizing the existence of an Hamiltonian Cycle.

> **INPUT :** A (complete) graph G, a cost function $w : E(G) \mapsto \mathbb{R}^+$
> **OUTPUT :** A spanning cycle $C \subseteq G$ of minimum cost.

It can be defined as a Linear Program by identifying a special vertex $v^* \in V(G)$ and searching a 2-regular graph in G whose trace in $G - v^*$ is acyclic. The formulation is given on page page 85.

Steiner Tree

Given a set S of vertices of a graph G, a Steiner Tree is a minimum subgraph graph connecting all the vertices of S together.

> **INPUT :** A graph G, a set S of vertices
> **OUTPUT :** A tree $T \subseteq G$ containing the vertices in S with minimum size.

It is also a acyclic graph T of minimum cost containing the vertices of S. See page 86 for the corresponding formulation.

Minor Testing

A graph H is a minor of a graph G is it can be obtained from G by successive edge deletions or edge contractions. Among Robertson and Seymour's results on graph minors [137] is a cubic time algorithm for the H-minor problem.

INPUT : A graph G (H is fixed)
OUTPUT : Existence of a H minor in G

Since, this algorithm seems to have been improved to quadratic time by Kawarabayashi, Kobayashi and Reed [101] (submitted), but so far the known algorithms are still highly enumerative, and despite their kind asymptotic behavior none has been addressed with a working implementation.

A H-minor in G can also be seen as a family $\{S_1, \dots, S_{|H|}\} \subseteq 2^{V(G)}$ of vertex-disjoint sets such that each $G[S_i]$ is connected (they correspond to the sets of vertices which – once merged – will correspond to the vertices of H), with the additional property that if $uv \in E(H)$ there exists an edge between one vertex from S_u and one vertex from S_v. This describes the formulation given on pages 87-88.

Unfortunately, this formulation is in practice pretty slow.

Edge-disjoint spanning tree

The problem of finding k edge-disjoint spanning trees in a graph is polynomial, and one of the earliest consequences of the Matroid Union Theorem (see Schrijver [143]).

INPUT : A graph G, an integer k
OUTPUT : (Existence of) k edge-disjoint spanning trees of G

The Linear Program can be written as a coloring of the edges such that each color class is a spanning tree. See page 90 for the corresponding formulation.

Linear arboricity

The Linear Arboricity is an edge-covering problem presented page 43.

INPUT : A graph G, an integer k
OUTPUT : (Existence of) a family of k linear forests covering the edges of G

This Linear Program can be written by assigning a color class to each edge, while ensuring that each of the k different graphs has maximum degree 2 and is acyclic. See page 89 for the corresponding formulation.

Acyclic edge coloring

Acyclic edge coloring is a variant of edge coloring presented page 40.

INPUT : A graph G, an integer k
OUTPUT : (Existence of) an acyclic edge coloring of G with k colors

This Linear Program can be written as a proper edge coloring problem, to which are added constraints ensuring that the graphs induced by each pair of color classes is a forest. See p.91 for the corresponding formulation.

Chapter 6

Appendices

Longest Path

- Maximize :

$$\sum_{e \in E(G)} b_e$$

- Such that :
  - P is of maximum degree 2

$$\forall v \in V(G)$$

$$\sum_{\substack{e \in E(G)\\ e \sim v}} b_e \leq 2$$

  - The number of vertices in P is the number of its edges increased by 1

$$1 + \sum_{e \in E(G)} b_e = \sum_{v \in V(G)} b_v$$

  - The two endpoints of an edge in P are also in P

$$\forall uv = e \in E(G)$$

$$b_e \leq b_u \text{ and } b_e \leq b_v$$

  ◇ The graph P is acyclic :
    * $Mad(P)$ is less than $2 - \frac{2}{n}$

$$\forall v \in V(G), \sum_{\substack{e \in E(G)\\ e \sim v}} x_{e,v} \leq 1 - \frac{1}{n}$$

    * Each edge sends a flow of 1 to its endpoints if it is in P, 0 otherwise :

$$\forall e = uv \in E(G)$$

$$x_{e,u} + x_{e,v} = b_e$$

- Variables :
  - b_e is a boolean variable indicating whether the corresponding edge is in the longest path
  - b_v is a boolean variable indicating whether the corresponding vertex is in the longest path
  - $x_{e,v}$ is a real positive variable, representing the flow sent by an edge e to one of its endpoints v.

Traveling Salesman Problem

- Minimize :

$$\sum_{e \in E(G)} w(e) b_e$$

- Such that :

  - C is 2-regular

$$\forall v \in V(G)$$
$$\sum_{\substack{e \in E(G) \\ e \sim v}} b_e = 2$$

  ◇ The graph $C - v^*$ is acyclic :

    * $Mad(C - v^*)$ is less than $2 - \frac{2}{n}$

$$\forall v \in V(G) \backslash v^*, \sum_{\substack{e \in E(G) \\ e \sim v}} x_{e,v} \leq 1 - \frac{1}{n}$$

    * Each edge of $G - v^*$ sends a flow of 1 to its endpoints if it is in C, 0 otherwise :

$$\forall e = uv \in E(G \backslash v^*)$$

$$x_{e,u} + x_{e,v} = b_e$$

- Variables :

  - b_e is a boolean variable indicating whether the corresponding edge is in the longest path
  - b_v is a boolean variable indicating whether the corresponding vertex is in the longest path
  - $x_{e,v}$ is a real positive variable, representing the flow sent by an edge e to one of its endpoints v.

Steiner Tree

- Minimize :

$$\sum_{e \in E(G)} w(e) b_e$$

- Such that :
  - T contains the vertices from S

$$\forall v \in S$$

$$b_v = 1$$

  - The number of vertices in T is the number of its edges increased by 1

$$1 + \sum_{e \in E(G)} b_e = \sum_{v \in V(G)} b_v$$

  - The two endpoints of an edge in T are also in T

$$\forall uv = e \in E(G)$$

$$b_e \leq b_u \text{ and } b_e \leq b_v$$

  ◇ The graph T is acyclic :
    - ∗ $Mad(T)$ is less than $2 - \frac{2}{n}$

$$\forall v \in V(G), \sum_{\substack{e \in E(G) \\ e \sim v}} x_{e,v} \leq 1 - \frac{1}{n}$$

    - ∗ Each edge of G sends a flow of 1 to its endpoints if it is in T, 0 otherwise :

$$\forall e = uv \in E(G)$$

$$x_{e,u} + x_{e,v} = b_e$$

- Variables :
  - b_e is a boolean variable indicating whether the corresponding edge is in the longest path
  - b_v is a boolean variable indicating whether the corresponding vertex is in the longest path
  - $x_{e,v}$ is a real positive variable, representing the flow sent by an edge e to one of its endpoints v.

H-minor (part 1)

- Minimize : Nothing
- Such that :
  - A vertex of G represents at most one vertex of H

$$\forall v \in V(G)$$

$$\sum_{h \in V(H)} b_{v,h} \leq 1$$

  - A vertex of H has a non-empty set of representants

$$\forall h \in V(H)$$

$$\sum_{v \in V(G)} b_{v,h} \geq 1$$

  ◇ Each vertex set S_h is connected (it contains a spanning tree T_h) :
    * The number of edges in T_h is equal to $|S_h| - 1$:

$$\forall h \in V(H)$$

$$1 + \sum_{e \in E(H)} b_{e,h} = \sum_{v \in V(G)} b_{v,h}$$

    * An edge $uv = e$ can belong to T_h only if both its endpoints belong to S_h

$$\forall h \in V(H)$$

$$b_{e,h} \leq b_{u,h} \text{ and } b_{e,h} \leq b_{v,h}$$

    * $Mad(T_h)$ is less than $2 - \frac{2}{n}$

$$\forall h \in V(H), \forall v \in V(G), \sum_{\substack{e \in E(G) \\ e \sim v}} x_{e,v,h} \leq 1 - \frac{1}{n}$$

    * Each edge of G sends a flow of 1 to its endpoints if it is in T_h, 0 otherwise :

$$\forall h \in V(H), \forall e = uv \in E(G)$$

$$x_{e,u,h} + x_{e,v,h} = b_{e,h}$$

H-minor (part 2)

- Such that :

  ◇ For any edge $h_1h_2 \in E(H)$, there is at least one corresponding edge in G:

    * An edge $e \in E(G)$ can represent $h_1h_2 \in E(H)$ only if one of it has endpoints in both sets S_{h_1} and S_{h_2}

$$\forall v_1v_2 \in E(G), \forall h_1h_2 \in E(H)$$

$$2b_{v_1,v_2,h_1,h_2} \leq b_{v_1,h_1} + b_{v_2,h_2} \text{ and } 2b_{v_2,v_1,h_1,h_2} \leq b_{v_1,h_1} + b_{v_2,h_2}$$

    * There is at least one representant per edge in H

$$\forall h_1h_2 \in E(H)$$

$$\sum_{v_1v_2 \in E(G)} b_{v_1,v_2,h_1,h_2} + b_{v_2,v_1,h_1,h_2} \geq 1$$

- Variables :

  – $b_{v,h}$ is a boolean variable indicating whether the corresponding vertex belongs to S_h
  – $b_{e,h}$ is a boolean variable indicating whether the corresponding edge belongs to T_h
  – $x_{e,v,h}$ is a real positive variable, representing the flow sent by an edge $e \in T_h$ to one of its endpoints v.
  – x_{v_1,v_2,h_1,h_2} is a boolean variable indicating whether edge $v_1v_2 \in E(G)$ represents $h_1h_2 \in E(H)$.

Linear Arboricity

- Minimize : Nothing
- Such that :
  - The edges belong to one of K linear forests

$$\forall e \in E(G)$$

$$\sum_{k \in \{1,\ldots,K\}} b_{e,k} = 1$$

  - A vertex has at most 2 incident edges with the same color

$$\forall v \in V(G), \forall k \in \{1,\ldots,K\}$$

$$\sum_{\substack{uv=e\in E(G)\\ v\sim e}} b_{e,k} \leq 2$$

  - Each color class induces an acyclic graph. The following is to be understood for any k of the K color classes
    * An edge whose ends are colored with c_1 and c_2 sends a flow of 1, else 0.

$$\forall uv \in E(G)$$

$$e^k_{uv} + e^k_{vu} \geq b_{e,k}$$

    * A vertex can absorb at most $\frac{n-1}{n}$

$$\forall v \in V(G)$$

$$\sum_{u \in N_G(v)} e^k_{uv} \leq \frac{n-1}{n}$$

- Variables :
  - $b_{e,c}$ is a boolean variable indicating whether the corresponding edge belongs to forest k
  - e^k_{uv} is a positive variable indicating the flow sent to v by the edge uv in the graph induced by the vertices of forest k

Edge-disjoint spanning trees

- Minimize : Nothing
- Such that :
  - Each edge can belong to one of K trees
$$\forall e \in E(G)$$
$$\sum_{k\in\{1,\ldots,K\}} b_{e,k} = 1$$
  - A tree has $n-1$ edges
$$\forall k \in \{1,\ldots,K\}$$
$$\sum_{e\in E(G)} b_{e,k} = n-1$$
  - Each tree is acyclic
    * An edge sends a flow of 1 to its endpoints when it is in a tree
$$\forall e = uv \in E(G), \forall k \in \{1,\ldots,K\}$$
$$e^k_{uv} + e^k_{vu} \geq b_{e,k}$$
    * A vertex can absorb at most $\frac{n-1}{n}$
$$\forall v \in V(G), \forall k \in \{1,\ldots,K\}$$
$$\sum_{u\in N_G(v)} e^k_{uv} \leq \frac{n-1}{n}$$
- Variables :
  - $b_{e,k}$ is a boolean variable indicating whether the corresponding edge belongs to tree k.
  - e^k_{uv} is a positive variable indicating the weight of edge uv in the graph induced by the edges of class k.

Acyclic edge coloring

- Minimize : Nothing
- Such that :
  - The edges receive a color from a set $\mathcal{C}$

$$\forall e \in E(G)$$

$$\sum_{c \in C} b_{e,c} = 1$$

  - The coloring is proper

$$\forall v \in V(G), \forall c \in C$$

$$\sum_{\substack{e \in E(G) \\ v \sim e}} b_{e,c} \leq 1$$

  - Each pair of color classes induces an acyclic graph. The following is to be understood for all $c_1, c_2 \in \binom{\mathcal{C}}{2}$
    * An edge colored with c_1 or c_2 sends a flow of 1 to its endpoints, and 0 otherwise.

$$\forall uv \in E(G)$$

$$e_{uv}^{c_1c_2} + e_{vu}^{c_1c_2} \geq b_{e,c_1} + b_{e,c_2}$$

    * A vertex can absorb at most $\frac{n-1}{n}$

$$\forall v \in V(G)$$

$$\sum_{u \in N_G(v)} e_{uv}^{c_1c_2} \leq \frac{n-1}{n}$$

- Variables :
  - $b_{v,c}$ is a boolean variable indicating whether the corresponding edge is colored with c
  - $e_{uv}^{c_1c_2}$ is a positive variable indicating the flow sent to v by the edge uv in the graph induced by colors c_1 and c_2

Acyclic edge-colouring of planar graphs*

Manu Basavaraju[†] L. Sunil Chandran[‡] Nathann Cohen[§] Frédéric Havet[§]
Tobias Müller[¶]

May 10, 2011

Abstract

A proper edge-colouring with the property that every cycle contains edges of at least three distinct colours is called an *acyclic edge-colouring*. The *acyclic chromatic index* of a graph G, denoted $\chi'_a(G)$ is the minimum k such that G admits an *acyclic edge-colouring* with k colours. We conjecture that if G is planar and $\Delta(G)$ is large enough then $\chi'_a(G) = \Delta(G)$. We settle this conjecture for planar graphs with girth at least 5. We also show that $\chi'_a(G) \leq \Delta(G) + 12$ for all planar G, which improves a previous result by Fiedorowicz et al. [15].

1 Introduction

A proper edge-colouring with the property that every cycle contains edges of at least three distinct colours is called an *acyclic edge-colouring*. The *acyclic chromatic index* of a graph G, denoted $\chi'_a(G)$ is the minimum k such that G admits an *acyclic edge-colouring* with k colours. Fiamčik [12] and later Alon, Sudakov and Zaks [2] conjecture that $\Delta(G)+2$ colours are enough.

Conjecture 1 (Fiamčik [12]–Alon, Sudakov and Zaks [2]) *For every graph G, $\chi'_a(G) \leq \Delta(G)+2$.*

This conjecture would be tight as there are cases where more than $\Delta+1$ colours are needed. Consider for example a graph G on $2n$ vertices with at least $2n^2-2n+2$ edges. The union of two perfect matchings is a cycle factor and thus contains a cycle. Hence in an acyclic edge-colouring, at most one colour class contains n edges. Hence there are at least $1 + \left\lceil \frac{2n^2-3n+2}{n-1} \right\rceil = 2n+1$ colours. So $\chi'_a(G) \geq \Delta(G)+2$.

Clearly, every graph with maximum degree at most 2 has acyclic chromatic index at most 3. If $\Delta(G) \leq 3$ then its line-graph $L(G)$ has maximum degree at most 4. Thus by Burnstein's results [10]

*This work was achieved independently by two different teams: M. Basavaraju and L. S. Chandran, and N. Cohen, F. Havet and T. Müeller. It was combined based on the advice of the editor.

†Computer Science and Automation department, Indian Institute of Science, Bangalore- 560012, India. manu@csa.iisc.ernet.in; iammanu@gmail.com

‡Computer Science and Automation department, Indian Institute of Science, Bangalore- 560012, India. sunil@csa.iisc.ernet.in

§Projet Mascotte, I3S(CNRS, UNSA) and INRIA, 2004 route des lucioles, BP 93, 06902 Sophia-Antipolis Cedex, France. Nathann.Cohen@sophia.inria.fr ; fhavet@sophia.inria.fr. Partially supported by the ANR Blanc AGAPE.

¶Centrum voor Wiskunde en Informatica (CWI), Science Park 123, 1098 XG Amsterdam, The Netherlands. tobias@cwi.nl

$\chi_a(L(G)) \leq 5$ and so $\chi'_a(G) \leq 5$. So Conjecture 1 holds for $\Delta(G) \leq 3$. In 1980, Fiamčik [13] conjectured that K_4 is the only cubic graph requiring five colours in an acyclic edge-colouring (and actually gave an uncorrect proof of it). More generally, Alon, Sudakov and Zaks [2] conjectured that if G is a Δ-regular graph then $\chi'_a(G) = \Delta + 1$ unless $G = K_{2n}$.

However as noted by Fiamčik [14], these two conjectures are false as $\chi'(K_{3,3}) = 5$. Improving a result of Basavaraju and Chandran [4] on non-regular subcubic connected graphs, Macajova and Mazák [19] proved that if G is subcubic and connected then $\chi'_a(G) \leq 4$ unless $G = K_4$ or $G = K_{3,3}$. Finally, Basavaraju, Chandran and Kummini [7] showed that all d-regular graphs with $2n$ vertices and $d > n$, require at least $d+2$ colors to be acyclically edge-coloured and for every odd n, $\chi'(K_{n,n}) = n+2$. They also showed that for every d,n such that $d \geq 5$, $n \geq 2d+3$ and dn even, there exist d-regular graphs which require at least $d+2$-colours to be acyclically edge-coloured.

Alon, Sudakov and Zaks [2] showed that Conjecture 1 is true for almost all regular graphs. This was later improved by Nešetřil and Wormald [23] who proved that the acyclic edge-chromatic number of a random Δ-regular graph is asymptotically almost surely equal to $\Delta + 1$. Alon, McDiarmid and Reed [1] showed an upper bound of $64\Delta(G)$ for $\chi'_a(G)$ which was later improved to $16\Delta(G)$ by Molloy and Reed [20]. For graphs with large girth, better upper bounds are known. Muthu et al [21] showed that if G has girth at least 9 then $\chi'_a(G) \leq 6\Delta(G)$ and if it has girth at least 220 then $\chi'_a(G) \leq 4.52\Delta(G)$. Finally, Alon, Sudakov and Saks also showed that Conjecture 1 is true for graphs with girth at least $C\Delta\log(\Delta)$ for some fixed constant C.

Muthu et al [22] proved that $\chi'_a(G) \leq \Delta(G) + 1$ for outerplanar graphs. Fiedorowicz et al. [15] proved that $\chi'_a(G) \leq 2\Delta(G) + 29$ if G is planar and $\chi'_a(G) \leq \Delta(G) + 6$ if G is planar and triangle-free. This bound has been improved for planar graphs with larger girth. Recall that the *girth* of a graph is the minimum length of a cycle it contains or $+\infty$ if it has no cycles. Hou et al. [17] showed that if G is a planar graph G then $\chi'_a(G) \leq \Delta(G) + 2$ if G has girth at least 5, $\chi'_a(G) \leq \Delta(G) + 1$ if G has girth at least 7 and $\chi'_a(G) \leq \Delta(G)$ if G has girth at least 16 and $\Delta(G) \geq 3$.

Sanders and Zhao [24] showed that planar graphs with maximum degree $\Delta \geq 7$ have chromatic index Δ. A conjecture of Vizing [25] asserts that planar graphs of maximum degree 6 are also 6-edge-colourable. This would be best possible as for any $\Delta \in \{2,3,4,5\}$, there are some planar graphs with maximum degree Δ with chromatic index $\Delta + 1$ [25].

We propose a conjecture analogous to the above one of Vizing.

Conjecture 2 There exists Δ_0 such that every planar graph with maximum degree $\Delta \geq \Delta_0$ has an acyclic edge-colouring with Δ colours.

In this paper, we give some evidences to this conjecture. Firstly, in Section 2, we show that every planar graph has an acyclic edge-colouring with $\Delta + 12$ colours thus improving the $2\Delta + 29$ bound of Fiedorowicz et al. [15]. In Section 3, we show that Conjecture 2 holds for planar graphs of girth at least 5 (with $\Delta_0 = 19$) thus improving the results of Hou et al. [17] and Borowiecki and Fiedorowicz [9]. More generally, we settle Conjecture 2 for graphs with maximum average degree less than $4 - \varepsilon$ for any $\varepsilon > 0$. The *maximum average degree* of G is $Mad(G) = \max\{\frac{2|E(H)|}{|V(H)|} \mid H \text{ is a subgraph of } G\}$. It is well known that a planar graph of girth g has maximum average degree less than $2 + \frac{4}{g-2}$. Conjecture 2 holds for outerplanar graphs with $\Delta_0 = 5$ as shown by Hou et al. [18]. Note that $\sup\{Mad(G) \mid G \text{ is outerplanar}\} = 4$.

Our proofs are constructive and yield efficient polynomial time algorithms. We present the proofs in a non-algorithmic way. But it is easy to extract the underlying algorithms from them.

2 Planar graphs

In this section we will prove the following result.

Theorem 3 $\chi'_a(G) \le \Delta(G) + 12$ *for all planar graphs G.*

The proof of Theorem 3 relies on the following theorem of van den Heuvel and McGuiness [16] which establishes a set of unavoidable configurations in planar graphs.

Lemma 4 (van den Heuvel and McGuiness [16]) *Let G be a planar graph with minimum degree at least two. Then there exists a vertex v in G with exactly $d(v) = k$ neighbours $v_1, v_2, \ldots, v_k$ with $d(v_1) \le d(v_2) \le \ldots \le d(v_k)$ such that at least one of the following is true:*

(A1) $k = 2$,

(A2) $k = 3$ *and* $d(v_1) \le 11$,

(A3) $k = 4$ *and* $d(v_1) \le 7$, $d(v_2) \le 11$,

(A4) $k = 5$ *and* $d(v_1) \le 6$, $d(v_2) \le 7$, $d(v_3) \le 11$.

Sketch of the proof: Let G be a minimum counter-example with respect to the number of vertices and edges for the statement in Theorem 3. Trivially G has minimum degree at least 2. Indeed, it has no vertex v of degree 1 because any acyclic edge-colouring of $G - v$ is an acyclic edge-colouring of G, and it has no vertex v with a unique neighbour u, since any acyclic edge-colouring of $G - v$ on at least Δ colours may be extended to an acyclic edge-colouring of G by assigning to uv a colour not already assigned to an edge incident to u. From Lemma 4, we know that there exists a vertex v in G such that it belongs to one of the configurations $A1$–$A4$. If there is a configuration A_2, A_3 and A_4 in G, we show in Subsection 2.2 how to derive an acyclic edge-colouring with $\Delta + 12$ colours of G from one of $G \setminus vv_1$. Hence, we assume that there is no such configurations. In such case, we select an appropriate edge uu' and show again how to derive an acyclic edge-colouring of G with $\Delta + 12$ colours from one of $G \setminus uu'$. This gives a final contradiction. See Subsection 2.3.

In order to show how to extend an acyclic edge-colouring of $G \setminus e$ for some edge e into an acyclic edge-colouring of G,we first establish some preliminaries.

2.1 Preliminaries

Partial edge-colouring: Let H be a subgraph of G. Then an edge-colouring c' of H is also a *partial edge-colouring* of G. Note that H can be G itself. Thus an edge-colouring c of G itself can be considered a partial edge-colouring. A partial edge-colouring c of G is said to be a *proper partial edge-colouring* if c is proper. A proper partial edge-colouring c is called *acyclic* if there are no bichromatic cycles in the graph. Note that with respect to a partial edge-colouring c, $c(e)$ may not be defined for an edge e. So, whenever we use $c(e)$, we are considering an edge e for which $c(e)$ is defined, though we may not always explicitly mention it.

Let c be a partial edge-colouring of G. We denote the set of colours in c by $C = \{1, 2, \ldots, k\}$. For any vertex $u \in V(G)$, we define $F_u(c) = \{c(uz) \mid z \in N_G(u)\}$. For an edge $ab \in E$, we define $S_{ab}(c) = F_b(c) - \{c(ab)\}$. Note that $S_{ab}(c)$ need not be the same as $S_{ba}(c)$. We will abbreviate the notation to F_u and S_{ab} when the edge-colouring c is understood from the context.

The following definitions arise out of our attempt to understand what may prevent us from extending a partial edge-colouring of $G \setminus e$ to G.

Maximal bichromatic Path: An *(α,β)-maximal bichromatic path* with respect to a partial edge-colouring c of G is a maximal path consisting of edges that are coloured using the colours α and β alternatingly. An *(α,β,a,b)-maximal bichromatic path* is an (α,β)-maximal bichromatic path which starts at the vertex a with an edge coloured α and ends at b. We emphasize that the edge of the (α,β,a,b)-maximal bichromatic path incident on vertex a is coloured α and the edge incident on vertex b can be coloured either α or β. Thus the notations (α,β,a,b) and (α,β,b,a) have different meanings. Also note that any maximal bichromatic path will have at least two edges. The following fact is obvious from the definition of proper edge-colouring.

Fact 5 *Given a pair of colours α and β of a proper edge-colouring c of G, there is at most one maximal (α,β)-bichromatic path containing a particular vertex v, with respect to c.*

A colour $\alpha \neq c(e)$ is a *candidate* for an edge e in G with respect to a partial edge-colouring c of G if none of the adjacent edges of e is coloured α. A candidate colour α is *valid* for an edge e if assigning the colour α to e does not result in any bichromatic cycle in G.

Let $e = ab$ be an edge in G. Note that any colour $\beta \notin F_a \cup F_b$ is a candidate colour for the edge ab in G with respect to the partial edge-colouring c of G. A sufficient condition for a candidate colour being valid is captured in the lemma below.

Lemma 6 (Basavaraju and Chandran [6]) *A candidate colour for an edge $e = ab$ is valid if $(F_a(c) \cap F_b(c)) \setminus \{c(ab)\} = S_{ab}(c) \cap S_{ba}(c) = \emptyset$.*

Now even if $S_{ab}(c) \cap S_{ba}(c) \neq \emptyset$, a candidate colour β may be valid. But if β is not valid, then what may be the reason? It is clear that colour β is not *valid* if and only if there exists $\alpha \neq \beta$ such that a (α,β)-bichromatic cycle gets formed if we assign colour β to the edge e. In other words, if and only if, with respect to edge-colouring c of G there existed an (α, β, a, b)-maximal bichromatic path with α being the colour given to the first and last edge of this path. Such paths play an important role in our proofs. We call them *critical paths*. It is formally defined below.

Critical Path: Let $ab \in E$ and c be a partial edge-colouring of G. Then an (α, β, a, b)-maximal bichromatic path which starts out from the vertex a via an edge coloured α and ends at the vertex b via an edge coloured α is called an *(α, β, a, b)-critical path*. Note that any critical path will be of odd length. Moreover the smallest length possible is three.

Let $a \in N_{G \setminus vv_1}(x)$ and let $c(x,a) = \alpha$. Let $\beta \in S_{xa}$. colour β is said to be *actively present* in a set S_{xa}, if there exists a (α, β, xy) critical path.

A natural strategy to extend a acyclic partial edge-colouring c of G would be to try to assign one of the candidate colours to an uncoloured edge e. The condition that a candidate colour being not valid for the edge e is captured in the following fact.

Fact 7 *Let c be a partial edge-colouring of G. A candidate colour β is not valid for the edge $e = ab$ if and only if for some colour $\alpha \in S_{ab} \cap S_{ba}$, there is an (α, β, a, b)-critical path in G with respect to c.*

Colour exchange: Let c be a partial edge-colouring of G. Let $u, v, w \in V(G)$ and $uv, uw \in E(G)$. We define *colour exchange* with respect to the edge uv and uw, as the modification of the current partial edge-colouring c by exchanging the colours of the edges uv and uw to get a partial edge-colouring c', i.e., $c'(uv) = c(uw)$, $c'(uw) = c(uv)$ and $c'(e) = c(e)$ for all other edges e in G. The colour exchange with respect to the edges uv and uw is said to be *proper* (resp. *acyclic*) if the edge-colouring obtained after the exchange is proper (resp. acyclic). The following fact is obvious.

Fact 8 *Let c' be the partial edge-colouring obtained from an acyclic partial edge-colouring c by the colour exchange with respect to the edges uv and uw. Then c' is proper if and only if $c(uv) \notin S_{uw}$ and $c(uw) \notin S_{uv}$.*

The colour exchange is useful in breaking some critical paths as is clear from the following lemma.

Lemma 9 (Basavaraju and Chandran [6, 5]) *Let u, v, w, a and b be vertices of G such that uv, uw and ab are edges. Also let α and β be two colours such that $\{\alpha,\beta\} \cap \{c(uv), c(uw)\} \neq \emptyset$ and $\{v,w\} \cap \{a,b\} = \emptyset$. Suppose there exists a (α,β,a,b)-critical path that contains vertex u, with respect to an acyclic partial edge-colouring c of G. Let c' be the partial edge-colouring obtained from c by the colour exchange with respect to the edges uv and uw. If c' is proper, then there is no (α,β,a,b)-critical path in G with respect to c'.*

Multisets and Multiset Operations: Recall that a *multiset* is a generalized set where a member can appear multiple times. If an element x appears t times in the multiset S, then we say that the *multiplicity* of x in S is t. In notation $mult_S(x) = t$. The cardinality of a finite multiset S, denoted by $\parallel S \parallel$, is defined as $\parallel S \parallel = \sum_{x \in S} mult_S(x)$. Let S_1 and S_2 be two multisets. The reader may note that there are various possible ways to define union of S_1 and S_2. For the purpose of this paper we define one such union notion- which we call as the *join* of S_1 and S_2, denoted as $S_1 \uplus S_2$. The multiset $S_1 \uplus S_2$ have all the members of S_1 as well as S_2. For a member $x \in S_1 \uplus S_2$, $mult_{S_1 \uplus S_2}(x) = mult_{S_1}(x) + mult_{S_2}(x)$. Clearly $\parallel S_1 \uplus S_2 \parallel = \parallel S_1 \parallel + \parallel S_2 \parallel$. We also need a specially defined notion of the multiset difference of S_1 and S_2, denoted by $S_1 \setminus S_2$. It is the multiset of elements of S_1 which are not in S_2, i.e., $x \in S_1 \setminus S_2$ iff $x \in S_1$ but $x \notin S_2$ and $mult_{S_1 \setminus S_2}(x) = mult_{S_1}(x)$.

2.2 There exists a Configuration $A2$, $A3$ or $A4$

We now can resume the proof of Theorem 3. Suppose by way of contradiction that there exists a Configuration A_2, A_3 or A_4 in G. Let v, v_1, v_2 and v_3 be the vertices as described in Lemma 4.

In all the propositions of this subsection, we start with an acyclic edge-colouring c' of $G \setminus vv_1$. So the abbreviations F_u and S_{ab} stand for $F_u(c')$ and $S_{ab}(c')$ respectively.

Proposition 10 *For any acyclic edge-colouring c' of $G \setminus vv_1$, $|F_v \cap F_{v_1}| \geq 2$.*

Proof. Suppose by way of contradiction that there is an acyclic edge-colouring c' of $G \setminus vv_1$ with a set C of $\Delta + 12$ colours such that $|F_v \cap F_{v_1}| \leq 1$.

Assume first that $|F_v \cap F_{v_1}| = 0$. The reader can verify from close examination of Configurations $A2$, $A3$ and $A4$ that $|F_v \cup F_{v_1}|$ will be maximum for Configuration $A2$ and therefore $|F_v \cup F_{v_1}| = |F_v| + |F_{v_1}| \leq 2 + 10 = 12$. Thus there are Δ candidate colours for the edge vv_1 and by Lemma 6 all the candidate colours are valid, a contradiction to the assumption that G is a counter-example.

Assume now that $|F_v \cap F_{v_1}| = 1$. It is easy to see that $|F_v \cup F_{v_1}| = |F_v| + |F_{v_1}| - |F_v \cap F_{v_1}| \leq 11$ and hence there are at least $\Delta + 1$ candidate colours for the edge vv_1. Let $F_v \cap F_{v_1} = \{\alpha\}$ and let $u \in N(v)$ be a vertex such that $c'(vu) = \alpha$. Now if none of the $\Delta + 1$ candidate colours is valid for the edge vv_1, then by Fact 7, for each $\gamma \in C \setminus (F_v \cup F_{v_1})$, there exists an (α, γ, v, v_1)-critical path. Since $c'(vu) = \alpha$, we have all the critical paths passing through the vertex u and hence $S_{vu} \subseteq C \setminus (F_v \cup F_{v_1})$. This implies that $|S_{vu}| \geq |C \setminus (F_v \cup F_{v_1})| \geq (\Delta + 12) - 11 = \Delta + 1$, a contradiction since $|S_{vu}| \leq \Delta - 1$. Thus we have a valid colour for the edge vv_1, a contradiction to the assumption that G is a counter-example. □

Let S_v be the multiset defined by $S_v = S_{vv_2} \uplus S_{vv_3} \uplus \ldots \uplus S_{vv_k}$.

Proposition 11 *For any acyclic edge-colouring c' of $G \setminus vv_1$, $|F_v \cap F_{v_1}| \neq 2$.*

Proof. Suppose not. Let $F_v \cap F_{v_1} = \{\alpha_1, \alpha_2\}$ and let $v', v'' \in N_{G\setminus vv_1}(v)$ and $u', u'' \in N_{G\setminus vv_1}(v_1)$ be such that $c'(vv') = c'(v_1u') = \alpha_1$ and $c'(vv'') = c'(v_1u'') = \alpha_2$. It is easy to see that $|F_v \cup F_{v_1}| \leq 10$. Thus there are at least $\Delta+2$ candidate colours for the edge vv_1. If any of the candidate colours is valid for the edge vv_1, we are done. Thus none of the candidate colours is valid for the edge vv_1. This implies that there exists a $(\alpha_1, \theta, v, v_1)$- or $(\alpha_2, \theta, v, v_1)$-critical path for each candidate colour θ.

Claim 11.1 *The multiset S_v contains at least $|F_{v_1}|-1$ colours from F_{v_1}.*

Proof. Suppose not. Then there are at least two colours in F_{v_1} which are not in S_v. Let ν and μ be any two such colours. Now assign colours ν and μ to the edges vv' and vv'' respectively to get an edge-colouring c''. Now since $\nu, \mu \notin S_v$, we have $\nu \notin S_{vv'}$ and $\mu \notin S_{vv''}$. Moreover $\mu, \nu \notin \{\alpha_1, \alpha_2\}$. Thus the edge-colouring c'' is proper. Now we claim that the edge-colouring c'' is acyclic also. Suppose not. Then there has to be a bichromatic cycle containing at least one of the colours ν and μ. Clearly this cannot be a (ν, μ)-bichromatic cycle since $\mu \notin S_{vv'}$. Therefore it has to be a (ν, λ)- or (μ, λ)-bichromatic cycle where $\lambda \in F_v(c'') \setminus \{\nu, \mu\}$. Let u be a vertex such that $c''(vu) = \lambda$. This means that there was already a (λ, ν, v, v')- or (λ, μ, v, v'')-critical path with respect to the edge-colouring c'. This implies that $\nu \in S_{vu}$ or $\mu \in S_{vu}$, implying that $\nu \in S_v$ or $\mu \in S_v$, a contradiction. Thus the edge-colouring c'' is acyclic. Let $u_1, u_2 \in N_{G\setminus vv_1}(v_1)$ be such that $c''(v_1u_1) = \nu$ and $c''(v_1u_2) = \mu$.

Note that $|F_v \cup F_{v_1}| \leq 10$ (The maximum value of $|F_v \cup F_{v_1}|$ is attained when the graph has Configuration $A2$). Therefore there are at least $\Delta+2$ candidate colours for the edge vv_1. If any of the candidate colours are valid for the edge vv_1, then we are done as this is a contradiction to the assumption that G is a counter-example. Thus none of the candidate colours is valid for the edge vv_1 and therefore there exist either a (ν, θ, v, v_1)-critical or a (μ, θ, v, v_1)-critical path for each candidate colour θ. Let C_ν and C_μ respectively be the set of candidate colours which are forming critical paths with colours ν and μ. Then clearly $C_\nu \subseteq S_{v_1u_1}$ and $C_\mu \subseteq S_{v_1u_2}$ since $c''(v_1u_1) = \nu$ and $c''(v_1u_2) = \mu$. Now we exchange the colours of the edges vv' and vv'' to get a modified edge-colouring c. Note that c is proper since $\mu \notin S_{vv'}$ and $\nu \notin S_{vv''}$. By Lemma 9, all (ν, β, v, v_1)-critical paths where $\beta \in C_\nu$ and all (μ, γ, v, v_1)-critical paths where $\gamma \in C_\mu$ are broken. Now if none of the colours in C_ν are valid for edge vv_1, then it means that for each $\beta \in C_\nu$, there exists a (μ, β, v, v_1)-critical path with respect to the edge-colouring c, implying that $C_\nu \subseteq S_{v_1u_2}$. Since the recolouring involved no candidate colours, we still have $C_\mu \subseteq S_{v_1u_2}$. Thus we have $(C_\nu \cup C_\mu) \subseteq S_{v_1u_2}$. But $|C_\nu \cup C_\mu| \geq \Delta+2$ which implies that $|S_{v_1u_2}| \geq \Delta+2$, a contradiction since $|S_{v_1u_2}| \leq \Delta-1$. $\square$

Claim 11.2 *There exists at least two colours β_1 and β_2 in $C \setminus F_{v_1}$ with multiplicity at most one in S_v.*

Proof. In view of Claim 11.1 we have $\sum_{x \in C\setminus F_v} mult_{S_v}(x) = \| S_v \| - (|F_v|-1)$. Thus if $\| S_v \| - (|F_{v_1}| - 1) \leq 2|(C \setminus F_{v_1})| - 3$, then there exist at least two colours β_1 and β_2 in $C \setminus F_{v_1}$ with multiplicity at most one in S_v. Thus it is enough to prove $\| S_v \| \leq 2|C| - |F_{v_1}| - 4 \leq 2\Delta + 24 - |F_{v_1}| - 4 = 2\Delta + 20 - |F_{v_1}|$. Now we can easily verify that $\| S_v \| + |F_{v_1}| \leq 2\Delta + 20$ for Configurations $A2$, $A3$ and $A4$ as follows:

- For $A2$, $\| S_v \| + |F_{v_1}| \leq (d(v_2)-1) + (d(v_3)-1) + |F_{v_1}| = (\Delta-1) + (\Delta-1) \;\; + 10 = 2\Delta + 8$.
- For $A3$, $\| S_v \| + |F_{v_1}| \leq (d(v_2)-1) + (d(v_3)-1) + (d(v_4)-1) + |F_{v_1}| = 10 + (\Delta-1) + (\Delta-1) \;\; + 6 = 2\Delta + 14$.

- For $A4$, $\parallel S_v \parallel + |F_{v_1}| \leq (d(v_2)-1)+(d(v_3)-1)+(d(v_4)-1)+(d(v_5)-1)+|F_{v_1}| = 6+10+(\Delta-1)+(\Delta-1) \ + \ 5 = 2\Delta+19$.

□

The colours β_1 and β_2 of Claim 11.2 are crucial to the proof. Now we make another claim regarding β_1 and β_2:

Claim 11.3 β_1 *and* $\beta_2 \in F_v$.

Proof. Without loss of generality, let $\beta_1 \notin F_v$. Then recalling that $\beta_1 \notin F_{v_1}$, β_1 is a candidate for the edge vv_1. If it is not valid, then there exists either an $(\alpha_1, \beta_1, vv_1)$- or $(\alpha_2, \beta_1, v, v_1)$-critical path with respect to c'. Since the multiplicity of β_1 in S_v is at most one, we have the colour β_1 in exactly one of $S_{vv'}$ or $S_{vv''}$. Without loss of generality let $\beta_1 \in S_{vv''}$. Hence there exists an $(\alpha_2, \beta_1, v, v_1)$-critical path with respect to c'.

Now recolour the edge vv' with colour β_1 to get an edge-colouring c. Then c is proper since $\beta_1 \notin F_v$ and $\beta_1 \notin S_{vv'}$. We shall prove that is is acyclic. Suppose, by way of contradiction, that there is a bichromatic cycle with respect to c. Then it has to be a (β_1, γ)-bichromatic cycle for some $\gamma \in F_v(c) \setminus c(vv')$. Let $a \in N_{G \setminus vv_1}(v)$ be such that $c(va) = \gamma$. Then the (β_1, γ)-bichromatic cycle should contain the edge va and therefore $\gamma \in S_{va}(c)$. But we know that v'' is the only vertex in $N_{G\setminus vv_1}(v)$ such that $\beta_1 \in S_{vv''}$. Therefore $a = v''$. This implies that $\gamma = \alpha_2$ and there existed an $(\alpha_2, \beta_1, v, v')$-critical path with respect to the edge-colouring c'. This is a contradiction to Fact 5 since there already existed an $(\alpha_2, \beta_1, v, v_1)$-critical path with respect to the edge-colouring c'.

Thus the edge-colouring c is acyclic and $|F_v(c) \cap F_{v_1}(c)| = 1$, a contradiction to Proposition 10. □

Note that $\{\beta_1, \beta_2\} \cap \{\alpha_1, \alpha_2\} = \emptyset$ since $\beta_1, \beta_2 \notin F_{v_1}$. In view of Claim 11.3, we have $\{\alpha_1, \alpha_2, \beta_1, \beta_2\} \subseteq F_v$ and thus $|F_v| \geq 4$, which implies that $d(v) \geq 5$. Thus the vertex v belongs to Configuration $A4$. Therefore $d(v) = 5$ and $F_v = \{\alpha_1, \alpha_2, \beta_1, \beta_2\}$. There are at least $\Delta + 12 - (5+4-2) = \Delta + 5$ candidate colours for the edge vv_1. Also recall that $d(v_2) \leq 7$, $c'(vv') = c'(v_1u') = \alpha_1$ and $c'(vv'') = c'(v_1u'') = \alpha_2$.

Claim 11.4 $v_2 \notin \{v', v''\}$.

Proof. Suppose not. Then, without loss of generality, $v_2 = v'$ and $c'(vv_2) = \alpha_1$. Now if none of the $\Delta+5$ candidate colours is valid for the edge vv_1, then they all are in critical paths that contain either the edge vv' or the edge vv''. Now $|S_{vv'}| + |S_{vv''}| \leq 6 + \Delta - 1 = \Delta + 5$. Since each of the $\Delta+5$ candidate colours has to be present in either in $S_{vv'}$ or $S_{vv''}$, we infer that $S_{vv''} \cup S_{vv'}$ is exactly the set of candidate colours, i.e., $|S_{vv'}| + |S_{vv''}| = \Delta + 5$. This requires that $|S_{vv'}| = 6$, $|S_{vv''}| = \Delta - 1$ and $S_{vv''} \cap S_{vv'} = \emptyset$. Since for each $\gamma \in S_{vv''}$, we have $(\alpha_2, \gamma, v, v_1)$-critical path containing u'', we can infer that $S_{vv''} \subseteq S_{v_1u''}$ (Recall that $c'(v_1u'') = \alpha_2$). But since $|S_{v_1u''}| \leq \Delta - 1$, we have $S_{vv''} = S_{v_1u''}$. Thus $S_{v_1u''} \cap S_{vv'} = S_{vv''} \cap S_{vv'} = \emptyset$.

Now we exchange the colours of the edges vv' and vv'' to get an edge-colouring c. Hence $c(vv') = \alpha_2$ and $c(vv'') = \alpha_1$. The edge-colouring c is proper since $\alpha_2 \notin S_{vv'}$ and $\alpha_1 \notin S_{vv''}$ (Recall that $S_{vv'}$ and $S_{vv''}$ contain only candidate colours). We shall prove that c is also acyclic: A bichromatic cycle with respect to c has to be an (α_1, η)- or (α_2, η)-bichromatic cycle for some $\eta \in F_v$. Clearly it cannot be an (α_1, α_2)-bichromatic cycle since $\alpha_1 \notin S_{vv'}(c)$ and therefore $\eta \in \{\beta_1, \beta_2\}$ (Recall that $F_v = \{\alpha_1, \alpha_2, \beta_1, \beta_2\}$). This implies that either β_1 or β_2 belongs to $S_{vv'} \cup S_{vv''}$. But we know that $S_{vv'} \cup S_{vv''}$ is exactly the set of candidate colours for the edge vv_1, a contradiction since $\beta_1, \beta_2 \in F_v$ cannot be candidate colours for the edge vv_1.

Therefore the edge-colouring c is acyclic. By Lemma 9, all the existing critical paths are broken. Now consider a colour $\gamma \in S_{vv'}$. If it is still not valid then there has to be a $(\alpha_2, \gamma, v, v_1)$-critical path since $c(vv') = \alpha_2$ and $\gamma \notin S_{vv''}(c)$. This implies that $\gamma \in S_{v_1u''}(c)$, a contradiction since $S_{v_1u''}(c) \cap S_{vv'}(c) = \emptyset$. Thus we have a valid colour for the edge vv_1, a contradiction to the assumption that G is a counter-example. $\square$

From Claim 11.4, we infer that $c'(vv_2) \notin F_v \cap F_{v_1}$ since $F_v \cap F_{v_1} = \{c'(vv'), c(vv'')\} = \{\alpha_1, \alpha_2\}$. Therefore we have $c(vv_2) \in \{\beta_1, \beta_2\}$ since $F_v = \{\alpha_1, \alpha_2, \beta_1, \beta_2\}$. Without loss of generality let $c(vv_2) = \beta_1$. We know that the colour β_2 can be in at most one of $S_{vv'}$ and $S_{vv''}$ by Claim 11.2. Now let v' be such that $\beta_2 \notin S_{vv'}$. Note that $C \setminus (S_{vv'} \cup F_v \cup F_{v_1}) \neq \emptyset$ since $|S_{vv'} \cup F_v \cup F_{v_1}| \leq \Delta - 1 + 4 + 5 - 2 = \Delta + 6$. Assign a colour $\theta \in C \setminus (S_{vv'} \cup F_v \cup F_{v_1})$ to the edge vv' to get an edge-colouring c''. Now $|F_v(c'') \cap F_{v_1}(c'')| = 1$ so by Proposition 10, it is not acyclic. Hence there is a bichromatic cycle with respect to c''. This bichromatic cycle should involve one of the colours α_2, β_1, β_2 along with θ. Since the bichromatic cycle contains a colour from $S_{vv'}$ and $\beta_2 \notin S_{vv'}$, it cannot be a (θ, β_2)-bichromatic cycle. Now with respect to the edge-colouring c', colour θ was not valid for the edge vv_1 implying that there existed a $(\alpha_1, \theta, v, v_1)$- or $(\alpha_2, \theta, v, v_1)$-critical path. But $(\alpha_1, \theta, v, v_1)$-critical path was not possible since $\theta \notin S_{vv'}$ by the choice of θ. Thus there existed an $(\alpha_2, \theta, v, v_1)$-critical path with respect to c'. Thus by Fact 5, there cannot be an $(\alpha_2, \theta, v, v')$-critical path with respect to c' and hence there cannot be an (α_2, θ)-bichromatic cycle in c'' formed due to the recolouring. Thus if there is a bichromatic cycle formed, then it has to be a (β_1, θ)-bichromatic cycle, which implies that $\beta_1 \in S_{vv'}$.

Now taking into account the fact that β_1 is in $S_{vv'}$ as well as F_v, we get $|S_{vv'} \cup F_v \cup F_{v_1}| \leq \Delta - 1 + 4 + 5 - 2 - 1 = \Delta + 5$ and therefore $|S_{vv'} \cup F_v \cup F_{v_1} \cup S_{vv_2}| \leq \Delta + 5 + 6 = \Delta + 11$. Thus $C \setminus (S_{vv'} \cup F_v \cup F_{v_1} \cup S_{vv_2}) \neq \emptyset$. Now recolour the edge vv' using a colour $\gamma \in C \setminus (S_{vv'} \cup F_v \cup F_{v_1} \cup S_{vv_2})$ to get an edge-colouring c. Clearly this edge-colouring is proper. It is also acyclic since if a bichromatic cycle gets formed it has to be a (β_1, γ) bichromatic cycle (Note that the (α_2, γ) and (β_2, γ) bichromatic cycles are argued out as before). But $\gamma \notin S_{vv_2}$, a contradiction. Thus the edge-colouring c is acyclic.

But $|F_v(c) \cap F_{v_1}(c)| = 1$, a contradiction to Proposition 10. This completes the proof of Proposition 11. $\square$

Proposition 12 *For any acyclic edge-colouring c' of $G \setminus vv_1$, $|F_v \cap F_{v_1}| \neq 3$.*

Proof. Suppose not. Let c' be an acyclic edge-colouring of $G \setminus vv_1$ such that $|F_v \cap F_{v_1}| = 3$. Then $|F_v| \geq 3$ and therefore $d(v) \geq 4$. Thus v belongs to either configuration $A3$ or $A4$. Let S'_v be the multiset defined by $S'_v = S_v \setminus (F_{v_1} \cup F_v$. Let $v', v'', v''' \in N_{G \setminus vv_1}(v)$ be such that $\{c(vv'), c(vv''), c(vv''')\} = F_v \cap F_{v_1}$. Also let $c(vv') = \alpha_1$, $c(vv'') = \alpha_2$ and $c(vv''') = \alpha_3$.

Claim 12.1 $\parallel S'_v \parallel \leq 2\Delta + 11$.

Proof. When $d(v) = 4$, it is clear that $\parallel S'_v \parallel \leq (d(v_2) - 1) + (d(v_3) - 1) + (d(v_4) - 1) \leq 10 + \Delta - 1 + \Delta - 1 = 2\Delta + 8$. On the other hand when $d(v) = 5$, try to recolour one of the edges vv', vv'', vv''' using a colour in $C \setminus (F_v \cup F_{v_1})$. There are $\Delta + 6$ colours in $C \setminus (F_v \cup F_{v_1})$. If any of these colours is valid for one of vv', vv'' or vv''', then recolouring this edge with this colour, we obtain an acyclic edge-colouring c'' satisfying $|F_v(c'') \cap F_{v_1}(c'')| = 2$. This contradicts Proposition 11. Hence there has to be a bichromatic cycle formed during each recolouring. Since such a bichromatic cycle has to be a (γ_1, γ_2)-bichromatic cycle where γ_1 is the colour used in the recolouring and $\gamma_2 \in F_v \setminus \{\gamma_1\}$, we infer that $S_{vv'}$, $S_{vv''}$ and $S_{vv'''}$ contain at least one colour from F_v. Thus we have $\parallel S'_v \parallel \leq \parallel S_v \parallel -3 \leq (d(v_2) - 1) + (d(v_3) - 1) + (d(v_4) - 1) + (d(v_5) - 1) - 3 \leq 6 + 10 + \Delta - 1 + \Delta - 1 - 3 = 2\Delta + 11$. $\square$

Claim 12.2 *There exists at least one colour $\beta \in C \setminus (F_v \cup F_{v_1})$ with multiplicity at most one in S'_v.*

Proof. Since v belongs to either configuration $A3$ or configuration $A4$, we have $|F_v \cup F_{v_1}| \leq 9-3=6$. Thus $|C \setminus (F_v \cup F_{v_1})| \leq \Delta+6$. By *Claim* 12.1 we have $\parallel S'_v \parallel \leq 2\Delta+11$ and from this it is easy to see that there exists at least one colour $\beta \in C \setminus (F_v \cup F_{v_1})$ with multiplicity at most one in S'_v. □

Note that $\beta \in C \setminus (F_v \cup F_{v_1})$, where β is the colour from Claim 12.2 is a candidate colour for the edge vv_1. If it is not valid then there has to be a (θ, β, v, v_1)-critical path, where $\theta \in \{\alpha_1, \alpha_2, \alpha_3\}$. By Claim 12.2, β can be present in at most one of $S_{vv'}$, $S_{vv''}$ and $S_{vv'''}$. Without loss of generality let $\beta \in S_{vv''}$. Thus there exists an $(\alpha_2, \beta, v, v_1)$-critical path with respect to the edge-colouring c'. Recolour the edge vv' using the colour β to get an edge-colouring c. Clearly c is proper since $\beta \notin S_{vv'}$ and $\beta \notin F_v$. Let us show that it is also acyclic. A bichromatic cycle (with respect to c) has to contain the colour β as well as a colour $\gamma \in F_v(c) \setminus \{\beta\}$. If $\gamma = c(vw)$, then $\beta \in S_{vw}$, for the (β, γ)-bichromatic cycle to get formed. But v'' is the only vertex in $N_{G \setminus vv_1}(v)$ such that $\beta \in S_{vv''}$. Thus $w = v''$, $\gamma = \alpha_2$ and the cycle is an (α_2, β)-bichromatic cycle. This means that there existed an (α_2, β, v, v')-critical path with respect to the edge-colouring c', a contradiction to Fact 5 since there already existed an $(\alpha_2, \beta, v, v_1)$-critical path with respect to the edge-colouring c'. Thus the edge-colouring c is acyclic.

But $|F_v(c) \cap F_{v_1}(c)| = 2$, a contradiction to Proposition 11. This completes the proof of Proposition 12. □

Proposition 13 *For any acyclic edge-colouring c' of $G \setminus vv_1$, $|F_v \cap F_{v_1}| \neq 4$.*

Proof. Suppose not. Let c' be an acyclic edge-colouring of $G \setminus vv_1$ such that $|F_v \cap F_{v_1}| = 4$. Then $|F_v| \geq 4$ and since $d(v) \leq 5$, we have $d(v) = 5$. Hence v belongs to Configuration $A4$. Let S'_v be the multiset defined by $S'_v = S_v \setminus (F_{v_1} \cup F_v)$. Also let $c(vv_2) = \alpha_1$, $c(vv_3) = \alpha_2$, $c(vv_4) = \alpha_3$ and $c(vv_5) = \alpha_4$.

Now try to recolour an edge incident on v with a candidate colour from $C \setminus (F_v \cup F_{v_1})$. If the obtained edge-colouring c'' is acyclic then $|F_v(c'') \cap F_{v_1}(c'')| = 3$, a contradiction to Proposition 12. Hence there has to be a bichromatic cycle created due to recolouring with one of the colours from F_v. This implies that $F_v \cap S'_v \neq \emptyset$. Thus we have $\parallel S'_v \parallel \leq \parallel S_v \parallel -1 \leq (d(v_2)-1) + (d(v_3)-1) + (d(v_4)-1) + (d(v_5)-1) \leq 6+10+\Delta-1+\Delta-1-1 = 2\Delta+13$. Now since there are $|C \setminus (F_v \cup F_{v_1})| \geq \Delta+12-(4+5-4) = \Delta+7$ candidate colours and $\parallel S'_v \parallel \leq 2\Delta+13$, it is easy to see that there exists at least one candidate colour β with multiplicity at most one in S'_v.

Note that $\beta \in C \setminus (F_v \cup F_{v_1})$ is a candidate colour for the edge vv_1. If it is not valid then there has to be a (θ, β, v, v_1)-critical path, where $\theta \in \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$. We know that β can be present in at most one of S_{vv_2}, S_{vv_3}, S_{vv_4} and S_{vv_5}. Without loss of generality let $\beta \in S_{vv_3}$. Thus there exists an $(\alpha_2, \beta, v, v_1)$-critical path with respect to the edge-colouring c'. Recolour the edge vv_2 using the colour β to get an edge-colouring c. Clearly c is proper since $\beta \notin S_{vv_2}$ and $\beta \notin F_v$. Let us now show that it is acyclic. A bichromatic cycle with respect to c has to contain the colour β as well as a colour $\gamma \in F_v(c) \setminus \{\beta\}$. If $\gamma = c(vw)$, then $\beta \in S_{vw}$, for the (β, γ) bichromatic cycle to get formed. But v_3 is the only vertex in $N_{G \setminus vv_1}(v)$ such that $\beta \in S_{vv_3}$. Thus $w = v_3$, $\gamma = \alpha_2$ and it has to be a (β, α_2) bichromatic cycle. This means that there existed an $(\alpha_2, \beta, v, v_2)$-critical path with respect to the edge-colouring c', a contradiction to Fact 5 since there already existed an $(\alpha_2, \beta, v, v_1)$-critical path with respect to the edge-colouring c'. Thus the edge-colouring c is acyclic.

But $|F_v(c) \cap F_{v_1}(c)| = 3$, a contradiction to Proposition 12. □

By Lemma 4, $d_{G \setminus vv_1}(v) \leq 4$. Thus $|F_v \cap F_{v_1}| \leq |F_v| \leq 4$. Then Propositions 10, 11, 12 and 13 gives a contradiction to the assumption that G contains a Configuration $A2$, $A3$ or $A4$.

2.3 There is no Configuration $A2$, $A3$ or $A4$

In the previous subsection, we showed that G contains no Configuration $A2$, $A3$ or $A4$. Then by Lemma 4, there is a Configuration $A1$, that is a vertex v such that $d(v) = 2$. Now delete all the degree 2 vertices from G to get a graph H. Now since the graph H is also planar, there exists a vertex v' in H such that v' belongs to one of the configurations $A1$, $A2$, $A3$ or $A4$, say A'. The vertex v' was not already in Configuration A' in G. This means that the degree of at least one of the vertices of the configuration A' i.e., $\{v'\} \cup N_H(v')$, got decreased by the removal of 2-degree vertices. Let $P = \{x \in \{v'\} \cup N_H(v') : d_H(x) < d_G(x)\}$. Let u be the minimum degree vertex in P in the graph H. Now it is easy to see that $d_H(u) \leq 11$ since v' did not belong to A' in G.

Let $N'(u) = \{x | x \in N_G(u) \text{ and } d_G(u) = 2\}$. Let $N''(u) = N_G(u) - N'(u)$. It is obvious that $N''(u) = N_H(u)$.

Since $u \in P$ and $d_H(u) \leq 11$, we have $|N'(u)| \geq 1$ and $N''(u) \leq 11$. In G let $u' \in N'(u)$ be a two degree neighbour of u such that $N(u') = \{u, u''\}$. Now by minimality of G, the graph $G \setminus uu'$ admits an acyclic edge-colouring c' using a set C of $\Delta + 12$ colours. Let $F'_u = \{c'(ux) | x \in N'(u)\}$ and $F''_u = \{c'(ux) | x \in N''(u)\}$. Now if $c(u'u'') \notin F_u$ we are done since $|F_u \cup F_{u'}| \leq \Delta$ and thus there are at least 12 candidate colours which are also valid by Lemma 6.

We know that $|F''_v| \leq 11$. If $c'(u'u'') \in F'_v$, then let $c = c'$. Else if $c'(u'u'') \in F''_v$, then recolour edge $u'u''$ using a colour from $C \setminus (S_{u'u''} \cup F''_v)$ to get an edge-colouring c (Note that $|C \setminus (S_{u'u''} \cup F''_v)| \geq \Delta + 12 - (\Delta - 1 + 11) = 2$ and since u' has degree one in $G - \{uu'\}$, c is acyclic). Now if $c(u'u'') \notin F_u$ the proof is already discussed. Thus $c(u'u'') \in F'_u$.

Let us now consider the edge-colouring c. Let $a \in N'(u)$ be such that $c(ua) = c(u'u'') = \alpha$. Now if none of the candidate colours in $C \setminus (F_u \cup F_{u'})$ are valid for the edge uu', then by Fact 7, for each $\gamma \in C \setminus (F_u \cup F_{u'})$, there exists an (α, γ, u, u')-critical path. Since $c'(ua) = \alpha$, we have all the critical paths passing through the vertex a and hence $S_{ua} \subseteq C \setminus (F_u \cup F_{u'})$. This implies that $|S_{ua}| \geq |C \setminus (F_u \cup F_{u'})| \geq \Delta + 12 - (1 + \Delta - 1 - 1) = 13$, a contradiction since $|S_{ua}| = 1$. Thus we have a valid colour for the edge uu', a contradiction to the assumption that G is a counter-example.

This final contradiction completes the proof of Theorem 3.

3 Planar graphs of girth at least 5

The aim of this section is to prove Conjecture 2 for planar graphs of girth at least 5. Actually, we prove the conjecture for a more general class of graphs: the graphs of maximum average degree at most 10/3. The *average degree* of a graph G is $Ad(G) = \frac{1}{|V(G)|} \sum_{v \in V(G)} d(v) = \frac{2|E(G)|}{|V(G)|}$. The *maximum average degree* of G is $Mad(G) = \max\{Ad(H) \mid H \text{ is a subgraph of } G\}$. It is well known that the girth and the maximum average degree of a planar graph are related to each other:

Proposition 14 *Let G be a planar graph of girth g.*

$$Mad(G) < 2 + \frac{4}{g-2}.$$

Theorem 15 *Let $\Delta \geq 19$ and G be a graph with maximum degree at most Δ and maximum average degree less than $\frac{10}{3}$. Then $\chi'_a(G) \leq \Delta$.*

Theorem 15 and Proposition 14 immediately yield the following.

Corollary 16 *Let $\Delta \geq 19$ and G be a planar graph with maximum degree at most Δ and girth at least 5. Then $\chi'_a(G) \leq \Delta$.*

More generally than Theorem 15, we show the following.

Theorem 17 *For any $\varepsilon > 0$, there exists an integer Δ_ε such that every graph G with maximum degree at most Δ with $\Delta \geq \Delta_\varepsilon$ and maximum average degree less than $4-\varepsilon$ is acyclically Δ-edge-colourable.*

In order to prove Theorems 15 and 17, we first establish some properties of Δ-*minimal graphs* which are graphs with maximum degree at most Δ, not acyclically Δ-edge-colourable but such that every proper subgraph is. Then, by the Discharging Method, we deduce that such a graph has maximum average degree at least $4-\varepsilon$ (resp. $10/3$) if Δ is at least Δ_ε (resp. 22). We will first prove, in Subsection 3.2, Theorem 17 for its discharging procedure is simpler because we only establish the existence of Δ_ε and make no attempt to minimize it. We then show Theorem 15 in Subsection 3.3.

A vertex of degree i is called an *i-vertex* and an *i-neighour* of a vertex v is a neighbour of v having degree i.

3.1 Properties of Δ-minimal graphs

Proposition 18 *A Δ-minimal graph G is 2-connected. In particular, $\delta(G) \geq 2$.*

Proof. If G is not connected, it is the disjoint union of G_1 and G_2. Both G_1 and G_2 admits an acyclic Δ-edge-colouring by minimality of G. The union of these two edge-colourings is an acyclic Δ-edge-colouring of G.

Suppose now that G has a cutvertex v. Let C_i, for $1 \leq i \leq p$ be the components of $G-v$ and G_i the graph induced by $C_i \cup \{v\}$. By minimality of G, all the G_i admit an acyclic Δ-edge-colouring. Moreover, free to permute the colours we may assume that two edges incident to v get different colours. Hence the union of these edge-colourings is an acyclic Δ-edge-colouring of G because any cycle of G is entirely contained in one of the G_i. □

Proposition 19 *Let G be a Δ-minimal graph. For every vertex $v \in V(G)$, $\sum_{u\in N(v)} d(u) \geq \Delta+1$.*

Proof. Suppose by way of contradiction that there is a vertex v such that $\sum_{u\in N(v)} d(u) \leq \Delta$. Let w be a neighbour of v. By minimality of G, $G \setminus vw$ admits an acyclic edge-colouring with Δ colours. Now colour vw with a colour distinct from the ones of the edges incident to a neighbour of v. This is possible as there are at most $\Delta-1$ such edges distinct from vw. Doing so we clearly obtain a proper edge-colouring. Let us now show that there is no bicoloured cycle. A cycle that does not contain vw has edges of at least three colours as the edge-colouring of G was acyclic and a cycle containing vw must contain an edge vu and an edge tu with $u \in N(v) \setminus \{w\}$. By construction, the colours of tu, uv and vw are distinct. □

A *thread* is a path of length two whose internal vertex has degree 2.

Proposition 20 *Let $k \geq 2$ be an integer and G a Δ-minimal graph. In G, a Δ-vertex is the end of at most k threads whose other endvertex has degree at most k.*

To prove this proposition we need the following lemma.

Lemma 21 *Let $H = ((A,B),E)$ be a bipartite graph with $|A| = |B| = c$ such that for any vertex $a \in A$ $d(a) = 1$ and let $K_{A,B}$ be the complete bipartite graph with bipartition (A,B). If at least 3 vertices of B of degree at least one in H then there exists a perfect matching M of $K_{A,B}$ such that the bipartite graph $((A,B),E \cup M)$ has girth at least 6.*

Proof. Let m be the number of vertices of B of degree at least one. Let $b_1,\ldots,b_c$ be the vertices of B with $d(b_i) \geq 1$ if $i \leq m$ and $d(b_i) = 0$ otherwise. And let $a_1,\ldots,a_c$ be the vertices of A with $a_ib_i \in E$ for all $1 \leq i \leq m$. If $m \geq 3$, let $M = \{a_ib_{i+1} \mid 1 \leq i < m\} \cup \{a_mb_1\} \cup \{a_ib_i \mid m < i \leq c\}$. Then the unique cycle in $((A,B), E \cup M)$ is $C = (a_1,b_2,a_2,b_3,\ldots,a_{m-1},b_m,a_1)$. It has length $2m \geq 6$. □

Proof of Proposition 20. Suppose for a contradiction that there is a Δ-vertex u with $c = k+1$ threads uv_iw_i, $1 \leq i \leq c$, such $d(w_i) \leq k$. Note that $c \geq 3$.

Set $A = \{v_1,\ldots,v_c\}$. By Proposition 18, $w_i \notin A$ for all $1 \leq i \leq c$. By minimality of G, $G - A$ admits an acyclic Δ-edge-colouring.

Let us first extend it to the v_iw_i as follows. Let F be the set of colours assigned to the edges incident to u and to no vertex of A and for $1 \leq i \leq c$ let F_i be the set of colours assigned to the edges incident to w_i (and distinct from v_iw_i). Then $|F| = \Delta - c$ and $|F_i| \leq k-1$. For all $1 \leq i \leq c$, let S_i be the set of colours not in $F \cup F_i$. Since $|F| + |F_i| = \Delta - c + k - 1 = \Delta - 2$ then $|S_i| \geq 2$.

Assume first that $|\bigcup_{i=1}^{c} S_i| \geq 3$, then one can assign to each v_iw_i a colour in S_i in such a way that at least 3 colours appear on such edges and that different colours appear on v_iw_i and v_jw_j if$w_i = w_j$. We will now colour the edges uv_i for $1 \leq i \leq c$. Therefore let $H_1 = ((A,B), E_1)$ be the bipartite graph with B the set of c colours $\{b_1,\ldots,b_c\}$ not in F and in which v_i is adjacent to b_j if $c(v_iw_i) = b_j$. As long as some v_i has degree 0 then add an edge between a_i and an isolated b_j to obtain a bipartite graph $H_2 = ((A,B), E_2)$. Because at least three colours appear on the v_iw_i, the graph H_2 fulfils the hypothesis of Lemma 21. So there exists a perfect matching M of $K_{A,B}$ such that $((A,B), E_2 \cup M)$ has girth at least 6. For $1 \leq i \leq c$, assign to each uv_i the colour to which v_i is linked in M.

Let us now prove that this edge-colouring of G is acyclic. It is obvious that it is proper since v_i is not linked to $c(v_iw_i)$ in M. Let us now prove that it is acyclic. Let C be a cycle of G. If it contains no vertex of A then it contains edges of three different colours because the edge-colouring of $G - A$ is acyclic. Suppose now that C contains a unique vertex of A, say v_i. Then C contains w_iv_i, v_iu and ut with t a neighbour of u not in A. Then $c(ut) \in F$, so by construction, $c(w_iv_i) \neq c(ut)$. Hence the colours of w_iv_i, v_iu and ut are distinct. Suppose finally that C contains two vertices of A, say v_i and v_j. Then C contains w_iv_i, v_iu, w_jv_j and v_ju. Since $((A,B), E_2 \cup M)$ has girth at least 6, either $c(v_iu) \neq c(w_jv_j)$ or $c(v_ju) \neq c(w_iv_i)$. In both cases, C has edges of three different colours.

Asumme now that $|\bigcup_{i=1}^{c} S_i| < 3$. Then all the S_i are equal and of cardinality 2, say $S_i = \{a,b\}$ for all $1 \leq i \leq c$. Hence all the F_i are the same of cardinality $k-1$ and disjoint from F. Observe that this can happen only if all the w_i are distinct. Let us denote by $f_1,\ldots,, f_{k-1}\}$ the elements of the F_i. Let us set $c(v_iw_i) = a$ for $1 \leq i \leq k$, $c(v_cw_c) = b$, $c(uv_i) = f_i$ for $1 \leq i \leq k-1$, $c(uv_k) = b$ and $c(uv_{k+1} = a$. It is easy to check that the obtained edge-colouring is an acyclic edge-colouring of G.

□

Proposition 22 *Let k and l be two positive integers and G a Δ-minimal graph. In G, a $(\Delta - l)$-vertex is the end of at most $k - 1 - l$ threads whose other endvertex has degree at most k.*

To prove this proposition we need the following lemma.

Lemma 23 *Let $H = ((A,B), E)$ be a bipartite graph with $c = |A| < |B|$ such that for any vertex $a \in A$ $d(a) = 1$ and $K_{A,B}$ be the complete bipartite graph with bipartition (AB).*

Then there exists a perfect matching M of $K_{A,B}$ such that the bipartite graph $((A,B), E \cup M)$ has no cycle.

Proof. Let $c' = |B|$. Let $b_1, \ldots, b_{c'}$ be the vertices of B with $d(b_i) \geq 1$ if $i \leq m$ and $d(b_i) = 0$ otherwise. And let $a_1, \ldots, a_c$ be the vertices of A with $a_i b_i \in E$ for all $1 \leq i \leq m$. Let $M = \{a_i b_{i+1} \mid 1 \leq i \leq c\}$. This is well-defined since $c' > c$. Then $((A,B), E \cup M)$ has no cycle. □

Proof of Proposition 22. . Suppose for a contradiction that there is a $(\Delta - l)$-vertex u with $c = k - l$ threads uv_iw_i, $1 \leq i \leq c$, such $d(w_i) \leq k$.

Set $A = \{v_1, \ldots, v_c\}$. By minimality of G, $G - A$ admits an acyclic Δ-edge-colouring. Let us first extend it to the v_iw_i as follows. Let F be the set of colours assigned to the edges incident to u and to no vertex of A and for $1 \leq i \leq c$ let F_i be the set of colours assigned to the edges incident to w_i (and distinct from v_iw_i). Then $|F| = \Delta - l - c$ and $|F_i| \leq k - 1$.

For all $1 \leq i \leq c$ colour v_iw_i with a colour not in $F \cup F_i$ and distinct from the colours. This is possible since $|F| + |F_i| = \Delta - l - c + k - 1 = \Delta - 1$.

We will now colour the edges uv_i for $1 \leq i \leq c$. Therefore let $H_1 = ((A,B), E_1)$ be the bipartite graph with B the set of $c + j$ colours $\{b_1, \ldots, b_{c+j}\}$ not in F and in which v_i is adjacent to b_j if $c(v_iw_i) = b_j$. As long as some v_i has degree 0 then add an edge between a_i and an isolated b_j to obtain a bipartite graph $H_2 = ((A,B), E_2)$. Then H_2 fulfils the hypothesis of Lemma 23 so there exists a perfect matching M of $K_{A,B}$ such that $((A,B), E_2 \cup M)$ has no cycle. For $1 \leq i \leq c$, assign to each uv_i the colour to which v_i is linked in M.

In the same way as in the proof of Proposition 20, one shows that the obtained edge-colouring is acyclic. □

3.2 Proof of Theorem 17

Lemma 24 *Let $\varepsilon > 0$. There exists Δ_ε such that if $\Delta \geq \Delta_\varepsilon$ then any Δ-minimal graph has average degree at least $4 - \varepsilon$.*

Proof. The result for $\varepsilon = \frac{1}{2}$ implies the result for larger values of ε. Hence we assume that $\varepsilon \leq \frac{1}{2}$. Let us assign an initial charge of $d(v)$ to each vertex $v \in V(G)$ Set $d_\varepsilon = \left\lceil \frac{8}{\varepsilon} - 2 \right\rceil$.

We perform the following discharging rules.

R1: for $4 \leq d < d_\varepsilon$, every d-vertex sends $a(d) = 1 - \frac{4-\varepsilon}{d}$ to each neighbour.

R2: for $d_\varepsilon \leq d \leq \Delta + 1 - d_\varepsilon$ then every d-vertex sends $1 - \frac{\varepsilon}{2}$ to each neighbour.

R3: for $\Delta + 2 - d_\varepsilon \leq d \leq \Delta$ then every d-vertex sends

- $1 - \varepsilon$ to each 3-neighbour;
- $2 - \varepsilon$ to each 2-neighbour whose second neighbour has degree 2 or 3;
- $b(d) = 2 - \varepsilon - a(d)$ to each 2-neighbour whose second neighbour has degree d with $4 \leq d < d_\varepsilon$;
- $1 - \frac{\varepsilon}{2}$ to each 2-neighbour whose second neighbour has degree $d \geq d_\varepsilon$.

Let us now check that every vertex v has final charge $f(v)$ at least $4 - \varepsilon$.

If v is a 2-vertex then let u and w be its two neighbours with $d(u) \leq d(w)$. If $d(u) \leq 3$ then $d(w) \geq \Delta - 2$ by Proposition 19. Hence v receives $2 - \varepsilon$ from w by R3, so $f(v) \geq 2 + 2 - \varepsilon = 4 - \varepsilon$. If $4 \leq d(u) < d_\varepsilon$ then $d(w) > \Delta + 1 - d_\varepsilon$ by Proposition 19. Hence v receives $a(d)$ from u by R2 and

$b(d)$ from w by R3. So $f(v) = 4-\varepsilon$. If $d(u) \geq 10$ then v receives $1-\frac{\varepsilon}{2}$ from u and $1-\frac{\varepsilon}{2}$ from w by R3. So $f(v) = 4-\varepsilon$.

Suppose that v is a 3-vertex. Then by Proposition 19 it has at least two (≥ 8)-neighbours. Hence it receives at least $2 \times 1/2$ by R1, R2 or R3 because $\varepsilon \leq \frac{1}{2}$. So $f(v) \geq 4$.

Suppose $4 \leq d(v) < d_\varepsilon$. Then v sends $d(v)$ times $1-\frac{4-\varepsilon}{d(v)}$ so $f(v) \geq 4-\varepsilon$.

Suppose $d_\varepsilon \leq d(v) \leq \Delta+1-d_\varepsilon$. Then v sends at most $d(v)$ times $1-\frac{\varepsilon}{2}$ so $f(v) \geq d(v) \times \frac{\varepsilon}{2} \geq 4-\varepsilon$.

Suppose now that $d(v) \geq \Delta+2-d_\varepsilon$. Then by Propositions 20 and 22, the most v can send is when it has three 2-neighbours with second neighbour of degree at most 3, one 2-neighbour with second neighbour of degree d for all $4 \leq d \leq d_\varepsilon - 1$ and $\Delta - d_\varepsilon + 1$ 2-neighbours with second neighbour of degree at least d_ε. Hence

$$\begin{aligned} f(v) &\geq \Delta+2-d_\varepsilon-3(2-\varepsilon)-\sum_{d=4}^{d_\varepsilon-1} b(d)-(\Delta-d_\varepsilon+1)(1-\frac{\varepsilon}{2}) \\ &\geq \Delta\frac{\varepsilon}{2}-S_\varepsilon \end{aligned}$$

with $S_\varepsilon = d_\varepsilon - 2 + 3(2-\varepsilon) + \sum_{d=4}^{d_\varepsilon-1} b(d) - (1-\frac{\varepsilon}{2})(d_\varepsilon - 1)$. Setting $\Delta_\varepsilon = \left\lceil \frac{2}{\varepsilon}(S_\varepsilon + 4 - \varepsilon) \right\rceil$, if $\Delta \geq \Delta_\varepsilon$, $f(v) \geq 4-\varepsilon$. $\square$

Proof of Theorem 17. If Theorem 17 were false, then a minimum counterexample G would be a Δ-minimum graph. So by Lemma 24, its average degree would be at least $4-\varepsilon$, a contradiction. $\square$

3.3 Proof of Theorem 15

Lemma 25 *Let $\Delta \geq 19$ and G be a Δ-minimal graph. Then $Mad(G) \geq Ad(G) \geq 10/3$.*

Proof. Let us assign an initial charge of $d(v)$ to each vertex $v \in V(G)$ and perform the following discharging rules.

R1: every 4-vertex sends 4/9 to each of its (≤ 3)-neighbours;

R2: every 5-vertex sends 7/12 to each 2-neighbour and $1/3$ to each 3-neighbour;

R3: for $6 \leq d \leq 9$, every d-vertex sends $1-10/3d$ to each neighbour.

R4: for $10 \leq d \leq \Delta-9$ then every d-vertex sends $2/3$ to each neighbour.

R5: for $\Delta-8 \leq d \leq \Delta$ then every d-vertex sends

- 2/3 to each d-neighbour with $3 \leq d \leq 5$;
- 4/3 to each 2-neighbour whose second neighbour has degree 2 or 3;
- 8/9 to each 2-neighbour whose second neighbour has degree 4;
- 9/12 to each 2-neighbour whose second neighbour has degree 5;
- $1/3+10/3d$ to each 2-neighbour whose second neighbour has degree d with $6 \leq d \leq 9$;
- 2/3 to each 2-neighbour whose second neighbour has degree $d \geq 10$.

Let us now check that every vertex v has final charge $f(v)$ at least $\frac{10}{3}$.

If v is a 2-vertex then let u and w be its two neighbours with $d(u) \leq d(w)$. If $d(u) \leq 3$ then $d(w) \geq \Delta - 2$ by Proposition 19. Hence v receives 4/3 from w by R5, so $f(v) \geq 2 + 4/3 = 10/3$. If $d(u) = 4$ then $d(w) \geq \Delta - 3$ by Proposition 19. Hence v receives 4/9 from u by R1 and 8/9 from w by R5. So $f(v) = 10/3$. If $d(u) = 5$ then $d(w) \geq \Delta - 4$ by Proposition 19. Hence v receives 7/12 from u by R2 and 9/12 from w by R5. So $f(v) = 10/3$. If $6 \leq d(u) \leq 9$ then $d(w) \geq \Delta - 8$ by Proposition 19. Hence v receives $1 - 10/3d$ from u by R3 and $1/3 + 10/3d$ from w by R5. So $f(v) = 10/3$. If $d(u) \geq 10$ then v receives $2/3$ from u by R4 and $2/3$ from w by R5. So $f(v) = 10/3$.

Suppose that v is a 3-vertex. Then, since $\Delta \geq 10$, by Proposition 19 it has either a (≥ 5)-neighbour or two 4-neighbours. Hence it receives either at least 1/3 by R2, R3, R4 or R5, or $2 \times 4/9 \geq 1/3$ by R1. In both cases, $f(v) \geq 3 + 1/3 = 10/3$.

Suppose that v is a 4-vertex. Then, since $\Delta \geq 18$, by Proposition 19, it has either three (≤ 3)-neighbours and one (≥ 10)-neighbour or at most two (≤ 3)-neighbours. In the first case, it sends 4/9 to each of its 3-neighbours and receives 2/3 form its (≥ 10)-neighbour. So $f(v) \geq 4 - 3 \times \frac{4}{9} + \frac{2}{3} = 10/3$. In the second case, it sends 4/9 to at most 2 neighbours. So $f(v) \geq 4 - 2 \times \frac{4}{9} > 10/3$.

Suppose that v is a 5-vertex.
Assume first that v has at most three (≤ 3)-neighbours. If it has at least one (3)-neighbour it sends at most $3/2$ so $f(v) \geq 5 - 3/2 > 10/3$. If not it has three 2-neighbours. Let u_1 and u_2 be the two (≥ 4)-neighbours of v. By Proposition 19, $d(u_1) + d(u_2) \geq 11$ since $\Delta \geq 16$. Hence one of these two vertices is a (≥ 6)-vertex and it sends at least 4/9 to u. Hence $f(v) \geq 5 + 4/9 - 7/4 > 10/3$.
Assume now that v has at least four (≤ 3)-neighbours. Let i be the number of 2-neighbours of v. Then by Proposition 19, v has exactly $4 - i$ 3-neighbours and its fifth neighbour has degree at least $6 + i$ since $\Delta \geq 17$. Hence $f(v) \geq 5 - i.\frac{7}{12} - (4 - i)\frac{1}{3} + 1 - \frac{10}{3(6+i)} > 10/3$.

Suppose $6 \leq d(v) \leq 9$. Then v sends $d(v)$ times $1 - 10/3d(v)$ so $f(v) \geq d(v) - d(v)(1 - 10/3d) = 10/3$.

Suppose $10 \leq d(v) \leq \Delta - 10$. Then v sends at most $d(v)$ times $2/3$ so $f(v) \geq d(v)(1 - 2/3) \geq 10/3$.

Suppose that $d(v) = \Delta - l$ for $1 \leq l \leq 7$. By Proposition 22, v is incident to at most $\Delta - l - 1$ threads so its has at least one (≥ 3)-neighbour to which it sends at most $2/3$. Moreover the most it can send is when it has exactly one 2-neighbour with second neighbour of degree d for each $l + 2 \leq d \leq 9$ and $\Delta - 9$ 2-neighbours with second neighbour of degree at least 10. Hence its final charge is

$$
\begin{aligned}
f(v) &\geq \Delta - l - \left((\Delta - 8)\frac{2}{3} + \sum_{d=l+2}^{9} s(d) \right) \\
&\geq \frac{1}{3}\Delta + \frac{16}{3} - \left(l + \sum_{d=l+2}^{9} s(d) \right)
\end{aligned}
$$

with $s(3) = 4/3$, $s(4) = 8/9$, $s(5) = 9/12$ and $s(d) = 1/3 + 10/3d$ for $6 \le d \le 9$. Since $s(3) > 1$ and $s(d) < 1$ when $d \ge 4$, then $l + \sum_{d=l+2}^{9} s(d)$ is minimum when $l = 2$. Hence

$$\begin{aligned} f(v) &\ge \frac{1}{3}\Delta + \frac{16}{3} - \left(2 + \sum_{d=4}^{9} s(d)\right) \\ &\ge \frac{1}{3}\Delta + \frac{61}{36} - \frac{10}{3}\sum_{d=6}^{9}\frac{1}{d} \\ &\ge \frac{1}{3}\Delta + \frac{61}{36} - \frac{10}{3} \times \frac{275}{504} \ge \frac{10}{3} \end{aligned}$$

because $\Delta \ge 11$.

Suppose $d(v) = \Delta$. By Proposition 20, the most it can send is when it has three 2-neighbours with second neighbour of degree at most 3, exactly one 2-neighbour with second neighbour of degree d for $4 \le d \le 9$ and $\Delta - 9$ 2-neighbours with second neighbour of degree at least 10. In this case it sends

$$\begin{aligned} 3 \times \frac{4}{3} + \frac{8}{9} + \frac{9}{12} + \sum_{d=6}^{9}\left(\frac{1}{3} + \frac{10}{3d}\right) + (\Delta - 9)\frac{2}{3} &= \frac{2}{3}\Delta + \frac{35}{36} + \frac{10}{3}\sum_{d=6}^{9}\frac{1}{d} \\ &= \frac{2}{3}\Delta + \frac{35}{36} + \frac{10}{3} \times \frac{275}{504} \\ &\le \Delta - \frac{10}{3} \end{aligned}$$

because $\Delta \ge 19$. Hence $f(v) \ge \frac{10}{3}$.

Now $Ad(G) = \frac{1}{|V|}\sum_{v \in V(G)} d(v) = \frac{1}{|V|}\sum_{v \in V(G)} f(v) \ge \frac{10}{3}$. □

Proof of Theorem 15. If Theorem 15 would be false, a minimum counterexample G would be a Δ-minimum graph. So by Lemma 25, its average degree is at least $4 - \varepsilon$, a contradiction. □

4 Acknowledgement

We would like to thank Jan van den Heuvel for helpful discussions.

References

[1] N. Alon, C. McDiarmid and B. Reed. Acyclic colouring of graphs, *Random Structures and Algorithms* 2:277–288, 1991.

[2] N. Alon, B. Sudakov and A. Zaks. Acyclic edge-colorings of graphs. *J. Graph Theory* 37:157–167, 2001.

[3] N. Alon and A. Zaks. Algorithmic aspects of acyclic edge colorings. *Algorithmica* 32(4):611–614, 2002.

[4] M. Basavaraju and L. S. Chandran. Acyclic edge coloring of subcubic graphs. *Discrete Math.*, to appear.

[5] M. Basavaraju and S. Chandran. Acyclic edge coloring of 2-degenerate graphs. *Submitted*. Available at http://arxiv.org/abs/0803.2433v1, 2008.

[6] M. Basavaraju and S. Chandran. Acyclic edge coloring of graphs with maximum degree 4. *J. of Graph Theory* 61 (2009): 192–209, 2009.

[7] M. Basavaraju, L. S. Chandran and M. Kummini. d-regular graphs with acyclic chromatic index at least $d+2$. *J. Graph theory*, to appear.

[8] O.V. Borodin, H.J. Broersma, A. Glebov, and J. van den Heuvel. Minimal degrees and chromatic numbers of squares of planar graphs. (in Russian). *Diskretnyi Analiz i Issledovanie Operatsii. Seriya 1* :8(4), 9–33, 2001. English version with title Stars and bunches in planar graphs. Part II: General planar graphs and colourings available as CDAM Research Report CDAM-LSE-2002-05 `www.cdam.lse.ac.uk/Reports/Files/cdam-2002-05.pdf`

[9] M. Borowiecki and A. Fiedorowicz. Acyclic edge colouring of planar graphs without short cycles *Discrete Math.*, to appear.

[10] M. I. Burnstein. Every 4-valent graph has an acyclic 5-coloring (in Russian). *Soobšč. Akad. Nauk Gruzin. SSR* 93, 21–24, 1979.

[11] L. Esperet and P. Ochem. Oriented colorings of 2-outerplanar graphs. *Information Processing Letters*, 101(5):215–219, 2007.

[12] J. Fiamčik. The acyclic chromatic class of a graph (in russian). *Math. Slovaca* 28:139–145, 1978.

[13] J. Fiamčik. Acyclic chromatic index of a graph with maximum valency three. *Arch. Math. (Brno)* 16:81–88, 1980.

[14] J. Fiamčik. Acyclic chromatic index of subdivided graph. *Arch. Math. (Brno)* 20:69–82, 1984.

[15] A . Fiedorowicz, M. Haluszczak and N. Narayanan About acyclic edge colouring of planar graphs. *Information Processing Letters* 108(6):412-417, 2008.

[16] J. van den Heuvel and S. McGuinness. Coloring the square of a planar graph. *J. Graph Theory* 42:110–124, 2003.

[17] J. Hou, J. WU, G. Liu and B. Liu Acyclic edge colorings of planar graphs and series-parallel graphs *Science in China Series A: Mathematics* 51(3)–605–616, 2009.

[18] J.-F. Hou, J.-L. Wu, G.-Z. Liu and B. Liu. Acyclic edge chromatic number of outerplanar graphs. *J. Graph Theory* to appear, 2009.

[19] E. Máčajová and J. Mazák. Optimal acyclic edge-colourings of cubic graphs. *Technical Reports in Informatics, Comenius University, Bratislava* TR-2008-017, 2008.

[20] M. Molloy and B. Reed. Further algorithmic aspects of Lovasz Local Lemma. In *The 30th Annual ACM Symposium on Theory of Computing*, pp. 524–529, 1998.

[21] R. Muthu, N. Narayanan and C. R. Subramanian. Improved bounds on acyclic edge colouring. *Discrete Math.* 307(23):3063–3069, 2007.

[22] R. Muthu, N. Narayanan and C. R. Subramanian. Acyclic edge colouring of outerplanar graphs. *Lecture Notes in Computer Science* 4508:144–152, 2008.

[23] J. Nešetřil and N. C. Wormald. The acyclic edge chromatic number of a random d-regular graph is $d+1$. *J. Graph Theory* 49:69–74, 2005.

[24] D. P. Sanders and Y. Zhao. Planar graphs of maximum degree seven are class I. *J. Combin. Theory Ser. B* 83(2):201–212, 2001.

[25] V. G. Vizing. Critical graphs with given chromatic index. *Metody Diskret. Analiz* 5:9–17, 1965. [In Russian]

Linear and 2-frugal choosability of graphs of small maximum average degree

Nathann Cohen *　　Frédéric Havet *

May 30, 2011

Abstract

A proper vertex colouring of a graph G is *2-frugal* (resp. *linear*) if the graph induced by the vertices of any two colour classes is of maximum degree 2 (resp. is a forest of paths). A graph G is *2-frugally* (resp. *linearly*) *L-colourable* if for a given list assignment $L : V(G) \mapsto 2^{\mathbb{N}}$, there exists a 2-frugal (resp. linear) colouring c of G such that $c(v) \in L(v)$ for all $v \in V(G)$. If G is 2-frugally (resp. linearly) L-list colourable for any list assignment such that $|L(v)| \geq k$ for all $v \in V(G)$, then G is *2-frugally* (resp. *linearly*) *k-choosable*. In this paper, we improve some bounds on the 2-frugal choosability and linear choosability of graphs with small maximum average degree.

1 Introduction

The notion of acyclic colouring was introduced by Grünbaum [2] in 1973 : a vertex colouring is *acyclic* if it is *proper* (no two adjacent vertices have the same colour), and if there is no bicoloured cycle (the subgraph induced by the union of any two colours classes is a forest). A colouring such that for every vertex $v \in V(G)$, no colour appears more than p times in the neighborhood of v, is said to be *p-frugal*, a notion introduced by Hind, Molloy and Reed in [3]. The *p-frugal chromatic number* of a graph G, denoted by $\Phi_p(G)$, is the minimum number of colours in a p-frugal colouring of G and is clearly larger than $\lceil\frac{\Delta}{p}\rceil + 1$. Hind, Molloy, and Reed [3] proved that $\Phi_p(G) \leq \max(p\Delta(G), \frac{e^3}{p}\Delta(G)^{1+1/p})$. In addition, they show that this upper bound is tight up to within a constant factor by showing graphs G such that $\Phi_p(G) \geq \frac{1}{2p}\Delta(G)^{1+1/p}$.

Yuster [4] mixed the notions of 2-frugality and acyclicity, thus introducing the concept of linear colouring. A *linear colouring* of a graph is an acyclic and 2-frugal colouring. It can also be seen as a colouring such that the subgraph induced by the union of any two colour classes is a forest of paths (an acyclic graph with maximum degree at most two). The *linear chromatic number* of a graph G, denoted by $\Lambda(G)$, is the minimum number of colours in a linear colouring of G. As a linear colouring is 2-frugal, $\Lambda(G) \geq \Phi_2(G) \geq \lceil\frac{\Delta}{2}\rceil + 1$. Yuster proved in [4] that $\Lambda(G) = O(\Delta(G)^{3/2})$ in the general case, and he constructed graphs for which $\Lambda(G) = \Omega(\Delta(G)^{3/2})$.

These concepts may be generalized to list colouring. Given a list assignment $L : V(G) \mapsto 2^{\mathbb{N}}$, an *$L$-colouring* of G is a colouring c such that $c(v) \in L(v)$ for each vertex v. A graph G is *p-frugally* (resp. *linearly*) *L-colourable* if there is an L-colouring of G which is p-frugal (resp. linear). If G is p-frugally (resp. linearly) L-colourable for any assignment L verifying $\forall v \in V(G)$, $|L(v)| \geq k$, then G is said to be *p-frugally k-choosable* (resp. *linearly k-choosable*). The smallest integer k such that the graph G is p-frugally k-choosable is the *p-frugal choosability* or *p-frugal list chromatic number* of G and is denoted by $\Phi^l_p(G)$. The *linear choosability* or *linear list chromatic number* denoted $\Lambda^l(G)$ is defined analogously. The *average degree* of G is $Ad(G) = \frac{1}{|V(G)|}\sum_{v \in V(G)} d(v) = \frac{2|E(G)|}{|V(G)|}$. The *maximum average degree* of G is $Mad(G) = \max\{Ad(H) \mid H \text{ is a subgraph of } G\}$.

In [1], Esperet et al. proved some upper bounds on the linear choosability of graphs with small maximum average degree.

Theorem 1 (Esperet et al. [1])**.** *Let G be a graph with maximum degree at most Δ:*

1. *If $\Delta \geq 3$ and $Mad(G) < \frac{16}{7}$, then $\Lambda^l(G) = \lceil\frac{\Delta}{2}\rceil + 1$.*
2. *If $Mad(G) < \frac{5}{2}$, then $\Lambda^l(G) \leq \lceil\frac{\Delta}{2}\rceil + 2$.*
3. *If $Mad(G) < \frac{8}{3}$, then $\Lambda^l(G) \leq \lceil\frac{\Delta}{2}\rceil + 3$.*

*Projet Mascotte, CNRS/INRIA/UNSA, INRIA Sophia-Antipolis, 2004 route des Lucioles BP 93, 06902 Sophia-Antipolis Cedex, France .Partially supported by the ANR Blanc International Taiwan GRATEL. [Nathann.Cohen,fhavet]@sophia.inria.fr

In this article, we show in Theorem. 2 that the above upper bounds on the maximum average degree may be assumed arbitrarily close to 3 when Δ is large enough. When Δ is small, we also improve some of the existing bounds (see Theorem. 3). Since a linear colouring is 2-frugal, the results are also valid for 2-frugal choosability. However, since being 2-frugal is less restrictive than being linear, we improve some of them in this case (see Theorem. 4). All these results, added to those proved by Esperet et al. [1] which have not been improved, are summarized in the following table :

$Mad(G)$	Δ	$\Lambda^l(G)$	
$< \frac{16}{7} \approx 2.2857$	≥ 3	$\leq \lceil \frac{\Delta}{2} \rceil + 1$	Esperet et al. [1]
$< \frac{39}{16} = 2.4375$	≥ 5	$\leq \lceil \frac{\Delta}{2} \rceil + 1$	Theorem 3-1
$< \frac{48}{19} \approx 2.5263$	≥ 7	$\leq \lceil \frac{\Delta}{2} \rceil + 1$	Theorem 3-2
$< 3 - \frac{3}{\Delta+1}$	≥ 8	$\leq \lceil \frac{\Delta}{2} \rceil + 1$	Theorem 2-1
$< \frac{5}{2}$		$\leq \lceil \frac{\Delta}{2} \rceil + 2$	Esperet et al. [1]
$< \frac{60}{23} \approx 2.6086$	≥ 5	$\leq \lceil \frac{\Delta}{2} \rceil + 2$	Theorem 3-3
$< 3 - \frac{9}{4\Delta+3}$	≥ 7	$\leq \lceil \frac{\Delta}{2} \rceil + 2$	Theorem 2-2
$< \frac{14}{5} = 2.8$		$\leq \lceil \frac{\Delta}{2} \rceil + 3$	Theorem 3-4
< 3	≥ 12	$\leq \lceil \frac{\Delta}{2} \rceil + 3$	Theorem 2-3
< 3		$\leq \lceil \frac{\Delta}{2} \rceil + 4$	Theorem 3-5
$Mad(G)$	$\Phi^l_2(G)$	Δ	
$< \frac{5}{2}$	$\leq \lceil \frac{\Delta}{2} \rceil + 1$	≥ 7	Theorem 4-1
< 3	$\leq \lceil \frac{\Delta}{2} \rceil + 3$		Theorem 4-2

The *girth* $g(G)$ of a graph G is the length of its smallest cycle or $+\infty$ if G has no cycle. Euler's formula implies that a planar graph G has bounded maximum average degree in terms of its girth:

$$Mad(G) < 2 + \frac{4}{g(G)-2}. \tag{1}$$

This immediately gives to any result on graphs with bounded maximum average degree have an equivalent formulation for planar graphs with large girth. These are summarized in the following table together with those coming from the papers of Esperet et al. [1] and Raspaud and Wang [5, 6] which are not improved here.

girth	$\Lambda^l(G)$	Δ	
≥ 16	$\leq \lceil \frac{\Delta}{2} \rceil + 1$	≥ 3	Esperet et al. [1]
≥ 7	$\leq \lceil \frac{\Delta}{2} \rceil + 1$	≥ 13	Raspaud and Wang [5]
≥ 8	$\leq \lceil \frac{\Delta}{2} \rceil + 1$		
≥ 10	$\leq \lceil \frac{\Delta}{2} \rceil + 2$		Esperet et al
≥ 9	$\leq \lceil \frac{\Delta}{2} \rceil + 2$	≥ 5	
≥ 7	$\leq \lceil \frac{\Delta}{2} \rceil + 3$		
≥ 6	$\leq \lceil \frac{\Delta}{2} \rceil + 4$		
≥ 5	$\leq \lceil \frac{\Delta}{2} \rceil + 6$		Raspaud and Wang [6]
	$\leq \lceil \frac{9}{\Delta} 10 \rceil + 5$	≥ 85	Raspaud and Wang [6]
girth	$\Phi^l_2(G)$	Δ	
≥ 6	$\leq \lceil \frac{\Delta}{2} \rceil + 3$		

The proofs of our results are based on the same general idea. We study graphs which we call k*-frugal-minimal* (resp. k*-linear-minimal*) – i.e. graphs that are not k-frugally colourable (resp. k-linear-colourable), while any of their proper subgraphs is. We first show in Section 2 that some configurations (i.e. subgraphs) may not appear in such a graph. We then use in Section 3 the discharging method to show that a graph containing none of these forbidden configurations must have larger average degree than assumed, giving a contradiction .

2 Forbidden configurations

Before establishing some lemmas, let us give some definitions. Let k a non-negative integer. A *k-vertex* (resp. *$(\geq k)$-vertex*, *$(\leq k)$-vertex*) is a vertex of degree exactly k (resp. at least k, at most k). A *k-neighbour* of v is a k-vertex adjacent to v. $(\geq k)$ and $(\leq k)$-neighbours are defined similarly.
A *k-thread* in a graph G is an induced path of G with $k+1$ edges, and so k internal vertices of degree 2.

Note that a k-frugal-minimal or k-linear-minimal graph is connected and in particular has no 0-vertex. We will sometimes use this easy fact without referring explicitly to it.

2.1 Linear colouring

Lemma 1. *Let H be a k-linear-minimal or k-frugal-minimal graph.*

1. *If $k \geq \left\lceil \frac{\Delta(H)}{2} \right\rceil + 1$, then H has no 1-vertex.*
2. *For every 2-vertex v with $N(v) = \{a, b\}$ and $deg(a) \leq deg(b)$, we have $deg(b) \geq 2(k - deg(a)) + 1$.*
3. *If $k \geq 3$, then H contains no 3-thread.*
4. *If $k \geq 4$, then no 3-vertex is incident to a 2-thread.*
5. *Assume $k \geq 4$. If a 3-vertex has three 2-neighbours, then the second neighbour of each of those is a (≥ 4)-vertex.*
6. *If $k \geq 4$, then no 4-vertex is adjacent to four 2-threads.*
7. *If $k \geq 5$, then no 4-vertex is incident to a 2-thread.*
8. *If $k \geq 5$, then no 5-vertex is incident to five 2-threads.*
9. *If $k \geq 5$, then a 2-vertex has at most one 3-neighbour.*
10. *If $k \geq 5$, then a 4-vertex is not adjacent to two 2-vertices having each a 3-neighbour.*
11. *If $k \geq \left\lceil \frac{\Delta(H)}{2} \right\rceil + 2$, then two 2-vertices are not adjacent.*
12. *If $k \geq \left\lceil \frac{\Delta(H)}{2} \right\rceil + 3$, then every 3-vertex has no 2-neighbour.*
13. *If $k \geq \left\lceil \frac{\Delta(H)}{2} \right\rceil + 3$, then every 4-vertex has at most three 2-neighbours.*
14. *If $k \geq \lceil \frac{\Delta(H)}{2} \rceil + 4$, then every 4-vertex has at most four 2-neighbours.*
15. *If $k \geq \max\left(6, \lceil \frac{\Delta(H)}{2} \rceil + 4\right)$, then every 5-vertex has at most two 2-neighbours.*

Proof. In the following, we only prove the assertions for linear colouring : 2-frugal colouring being less restrictive, all the proofs translate naturally. Suppose that one of the assertions of Lemma 1 does not hold. Let H be a k-linear-minimal graph for which it fails and L a k-list assignment such that H is not linearly L-colourable.

1. H contains a 1-vertex u. Let v be the neighbour of u. Let c be a linear L-colouring of $H - u$. We now extend c to u. The colour $c(v)$ is forbidden. Moreover to preserve the 2-frugality at v, the colours appearing twice in the neighbourhood of v are also forbidden. There are at most $\lfloor \frac{\Delta(H)-1}{2} \rfloor$ such colours. Hence, at most $\lceil \frac{\Delta(H)}{2} \rceil$ colours in total are forbidden at u. Thus u can be coloured with a non-forbidden colour in its list $L(u)$, and the colouring obtained is a linear L-colouring of H, which is a contradiction.

2 Let v be a 2-vertex of H with $N(v) = \{a, b\}$ and $deg(a) \leq deg(b)$, such that $deg(b) < 2(k - deg(a)) + 1$. Let c be a linear L-colouring of $H - v$.

   – If $c(a) = c(b)$, let us assign to v a colour $c(v) \in L(v)$ different from the ones of the other neighbours of a (i.e. there are at most $deg(a) - 1$ of them), $c(a)$, and every colour which is repeated at least twice in the neighbourhood of b (i.e. there at most $\left\lfloor \frac{deg(b)-1}{2} \right\rfloor < k - deg(a)$ of them). Doing so, we obtain a k-linear L-colouring of H, a contradiction.

   – If $c(a) \neq c(b)$, let us assign to v a colour $c(v) \in L(v)$ different from $c(a)$, $c(b)$, and every colour which is repeated at least twice in the neighbourhood of a or in the neighbourhood of b. The number of forbidden colours is at most $2 + \left\lfloor \frac{deg(a)-1}{2} \right\rfloor + \left\lfloor \frac{deg(b)-1}{2} \right\rfloor < 2 + deg(a) - 2 + k - deg(a) = k$ because $deg(a) \geq 2$ and so $deg(a) \geq 2 + \left\lfloor \frac{deg(a)-1}{2} \right\rfloor$. Hence such an assignment is possible, and yields a k-linear L-colouring of H, a contradiction.

3. It follows directly from 2 which implies that if a 2-vertex has a 2-neighbour then its other neighbour has degree at least 3.

4. There is a configuration as depicted in Figure 1. with possibly $x = u_1$, or $x = u$ and $w = u_1$.

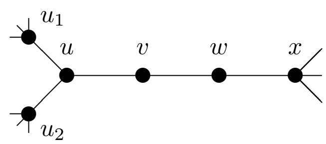


Figure 1: Configuration of Case 4

By the minimality of H, there exists a linear L-colouring c of $H - v$. We now extend it to v :

- If $c(u_1) = c(u_2)$, we colour v with $c(v) \in L(v)\backslash\{c(u), c(w), c(u_1)\}$. There can be no bicoloured cycle, as $c(v)$ is different from both $c(u_1)$ and $c(u_2)$, and the 2-frugality at u is preserved.
- If $c(u_1) \neq c(v_2)$, we colour v with $c(v) \in L(v)\backslash\{c(u), c(w), c(x)\}$. There can be no bicoloured cycle, as $c(v)$ is different from $c(x)$, and c is 2-frugal.

5. There is a configuration as depicted in Figure 2. Possibly $t_i = u_i$ and $w_i = u_i$ for $i = 1, 2$, or some vertices in $\{w_1, w_2, t_1, t_2\}$ are identified.

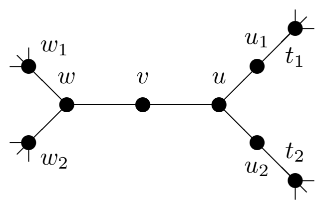


Figure 2: Configuration of Case 5

By the minimality of H there exists a linear L-colouring c of $H - \{u, v\}$. We now extend it to u and v :

- If $c(u_1) = c(u_2)$, we colour v with $c(v) \in L(v)\backslash\{c(w), c(w_1), c(u_1)\}$ and u with $c(u) \in L(v)\backslash\{c(v), c(u_1), c(t_1)\}$.
- If $c(u_1) \neq c(u_2)$, we colour u with $c(u) \in L(u)\backslash\{c(u_1), c(u_2), c(w)\}$ and v with $c(v) \in L(v)\backslash\{c(u), c(w), c(w_1)\}$.

6. There is a configuration as depicted in Figure 3. Possibly $t_3 = u = u_3$ and $u_1 = t_2$ and $t_1 = u_2$. In this case, a linear colouring of $H - \{u_1, u_2\}$ can be extended into a linear colouring of H by assigning to u_1 a colour distinct from those of u and w_1 and to u_2 a colour distinct from those of u, u_1 and v_1. This is a contradiction.

   So we may assume that it is not the case. Then possibly some vertices of $\{t_3, u_3, v_3, w_3\}$ may be identified.

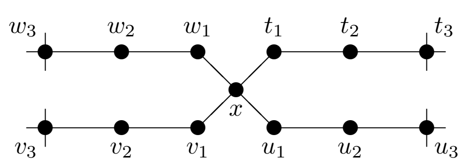


Figure 3: Configuration of Case 6

By the minimality of H there exists a linear L-colouring c of $H - \{t_1, u_1, v_1, w_1, x\}$. We now extend it to t_1, u_1, v_1, w_1 and x :

- We colour t_1 with $c(t_1) \in L(t_1)\backslash\{c(t_2), c(t_3)\}$.
- We colour u_1 with $c(u_1) \in L(u_1)\backslash\{c(u_2), c(u_3), c(t_1)\}$.
- We colour x with $c(x) \in L(x)\backslash\{c(t_1), c(u_1), c(w_2)\}$.
- We colour w_1 with $c(w_1) \in L(w_1)\backslash\{c(t_2), c(x)\}$.
- We colour v_1 with $c(v_1) \in L(v_1)\backslash\{c(v_2), c(x), c(w_1)\}$.

There can be no bicoloured cycle through t_1 (resp. u_1) as its colour is different from $c(t_3)$ (resp. $c(u_3)$), and none going through w_2 as its colour is different from $c(x)$.

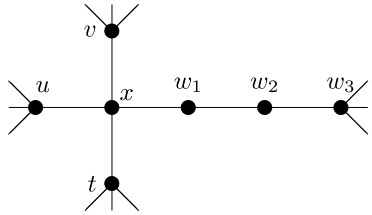


Figure 4: Configuration of Case 7

7. There is a configuration as depicted in Figure 4. Possibly w_3 is one of $\{t, u, v\}$ or $w_3 = x$ and $w_2 = t$. In the later case, we get a contradiction as in Case 6, so we may assume that it does not occur.

   By the minimality of H there exists a linear L-colouring c of $H - w_1$. We extend it to w_1 :

   – If $c(w_2) = c(x)$, then we colour w_1 with $c(w_1) \in L(w_1)\backslash\{c(x), c(w_3), c(v), c(t)\}$. There can be no bicoloured cycle as $c(w_1) \neq c(w_3)$, and the 2-frugality at x is preserved.
   – If $c(w_2) \neq c(x)$, then we colour w_1 with $c(w_1) \in L(w_1)\backslash\{c(x), c(w_2), c(v), c(t)\}$. There can be no bicoloured cycle as $c(w_2) \neq c(x)$, and the 2-frugality at x is preserved.

8. There is a configuration as depicted in Figure 5. Possibly, some of the vertices in $\{s_3, t_3, u_3, v_3, w_3\}$ are the same or identified with x (by pairs). In the later case, without loss of generality, $s_3 = x = t_3$, $s_1 = t_2$ and $t_1 = s_2$. Then, by minimality of H, there is a linear colouring c if $H - \{s_1, s_2\}$. It can be extended by colouring s_1 with a colour $c(s_1)$ distinct from $c(x)$ and the possible colour appearing twice on $\{u_1, v_1, w_1\}$, and colouring s_2 with a colour $c(s_2)$ distinct from $c(x)$, $c(s_1)$ and the possible colour appearing twice on $\{u_1, v_1, w_1\}$. Hence we may assume that this case does not appear.

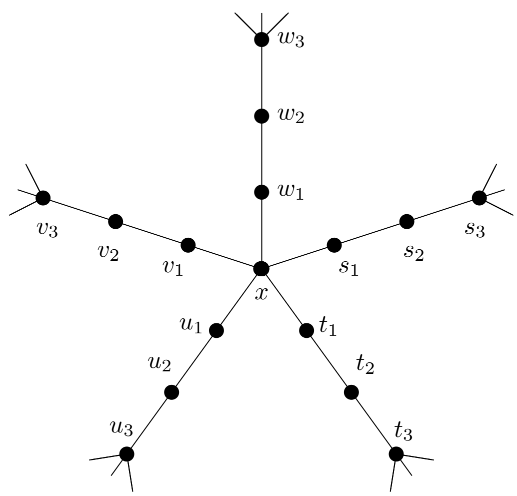


Figure 5: Configuration of Case 8

By the minimality of H there exists a linear L-colouring c of $H - \{x, s_1, t_1, u_1, v_1, w_1\}$. We extend it to x, s_1, t_1, u_1, v_1, and w_1 :

– We colour s_1 with $c(s_1) \in L(s_1)\backslash\{c(s_2), c(s_3)\}$.
– We colour t_1 with $c(t_1) \in L(t_1)\backslash\{c(t_2), c(t_3), c(s_1)\}$.
– We colour u_1 with $c(u_1) \in L(u_1)\backslash\{c(u_2), c(u_3), c(s_1), c(t_1)\}$.
– We colour v_1 with $c(v_1) \in L(v_1)\backslash\{c(v_2), c(u_1), c(t_1), c(s_1)\}$.
– We colour x with $c(x) \in L(x)\backslash\{c(s_1), c(t_1), c(u_1), c(v_1)\}$.
– We colour w_1 with $c(w_1) \in L(w)\backslash\{c(w_2), c(x), c(v_1)\}$.

The 2-frugality at x is preserved as 4 different colours are assigned to the vertices s_1, t_1, u_1 and v_1. Furthermore, there can be no bicoloured cycles going through s_1 and s_3, t_1 and t_3, u_1 and u_3 or w_1 and v_1. Thus the obtained L-colouring is linear, a contradiction.

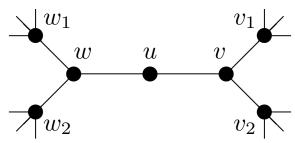


Figure 6: Configuration of Case 9

9. There is a configuration as depicted in Figure 6. Possibly, some of the vertices of $\{v_1, v_2, w_1, w_2\}$ are identified or $v_1 = v$ and $w_1 = w$.

   By the minimality of H, there exists a linear L-colouring c of $H - u$. We extend it to u :

   - If $c(v) = c(w)$, we colour u with $c(u) \in L(u) \backslash \{c(v), c(v_1), c(v_2), c(w_1)\}$ to prevent the formation of a bicoloured cycle and preserve the 2-frugality at v and w.
   - If $c(v) \neq c(w)$, we colour u with $c(u) \in L(u) \backslash \{c(v), c(v_1), c(w_1), c(w)\}$ to preserve the 2-frugality at v and w. There can be no bicoloured cycles because $c(v) \neq c(w)$.

   Hence H is linearly L-colourable, a contradiction.

10. There is a configuration as depicted in Figure 7. Possibly some vertices of $\{v_1'', v_1''', v_2'', v_2''', v_3, v_4\}$ are the same, or, for $i \in \{1, 2\}$, $v_i'' = u$ and $v_i' = v_{i+2}$.

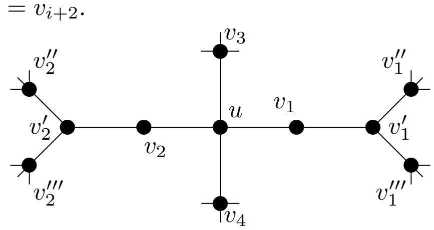


Figure 7: Configuration of Case 10

   By the minimality of H there exists a linear L-colouring c of $H - \{v_1, v_2\}$. We extend it to v_1 and v_2:

   - If $c(v_1') = c(u)$, we colour v_1 with $c(v_1) \in L(v_1) \backslash \{c(u), c(v_1''), c(v_1'''), c(v_4)\}$. There can be no bicoloured cycle through v_1 as it is different from both v_1'' and v_1''', and the 2-frugality at v_1' is preserved.
   - If $c(v_1') \neq c(u)$, we colour v_1 with $c(v_1) \in L(v_1) \backslash \{c(u), c(v_1'), c(v_1''), c(v_4)\}$. There can be no bicoloured cycle through v_1 as $c(v_1') \neq c(u)$, and the 2-frugality at v_1' is preserved.

   We colour v_2 with symmetrical rules, replacing v_4 by v_3.

   As $c(v_4) \neq c(v_1)$ and $c(v_3) \neq c(v_2)$, the 2-frugality of c is preserved.

11. It follows from 2. Indeed if a 2-vertex would have a 2-neighbour then its second neighbour has degree at least $\Delta(H) + 1$, a contradiction.

12. There is a configuration as depicted in Figure 8 with possibly $v_1' = v_2$.

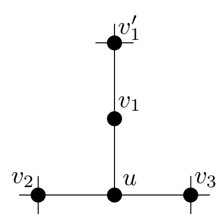


Figure 8: Configuration of Case 12

   By the minimality of H, there exists a linear L-colouring c of $H - v_1$.

   - If $c(u) = c(v_1')$, we colour v_1 with $c(v_1) \in L(v_1)$ that is different from $c(u)$, the colours appearing twice in the neighbourhood of v_1' (in order to preserve the 2-frugality at this vertex) and different from $c(v_2)$ and $c(v_3)$ to prevent the apparition of bicoloured cycles.

– If $c(u) \neq c(v_1')$, we colour v_1 with $c(v_1) \in L(v_1)$ that is different from $c(u)$ and $c(v_1')$, the colours appearing twice in the neighbourhood of v_1' (in order to preserve 2-frugality at this vertex) and different from $c(v_2)$ to preserve the 2-frugality at u. There can be no bicoloured cycles as $c(u) \neq c(v_1')$.

In both cases, H is linearly L-colourable, a contradiction.

13. There is a configuration as depicted in Figure 9 with possibly some of the vertices in $\{v_1', v_2', v_3', v_4'\}$ being identified.

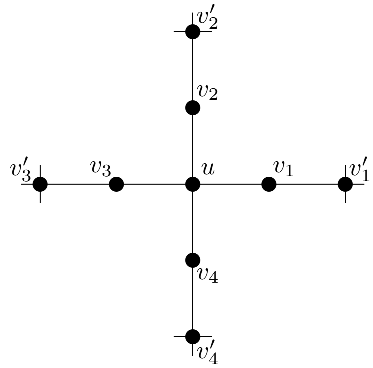


Figure 9: Configuration of Case 13

By the minimality of H there exists a linear L-colouring c of $H - \{u, v_1, v_2, v_3, v_4\}$. We extend it to u, v_1, v_2, v_3 and v_4 :

– We colour u with $c(u) \in L(u) \setminus \{c(v_1'), c(v_2'), c(v_3'), c(v_4')\}$ to prevent the apparition of bicoloured cycles.
– We colour v_1 with $c(v_1) \in L(v_1)$ different from $c(u)$, $c(v_1')$ and the colours appearing twice in the neighbourhood of v_1'.
– We colour v_2 with $c(v_2) \in L(v_2)$ different from $c(u)$, $c(v_2')$ and the colours appearing twice in the neighbourhood of v_2'.
– We colour v_3 with $c(v_3) \in L(v_3)$ different from $c(u)$, $c(v_3')$, $c(v_2)$ and the colours appearing twice in the neighbourhood of v_3'.
– We colour v_4 with $c(v_4) \in L(v_4)$ different from $c(u)$, $c(v_4')$, $c(v_1)$ and the colours appearing twice in the neighbourhood of v_4'.

There can be no bicoloured cycle with this colouring of H and c is 2-frugal because 3 vertices among $\{v_1, v_2, v_3, v_4\}$ cannot share the same colour. Then H is linearly L-colourable, a contradiction.

14. There is a configuration as depicted in Figure 10 with possibly some of the vertices in $\{v_1', v_2', v_3', v_4\}$ being identified.

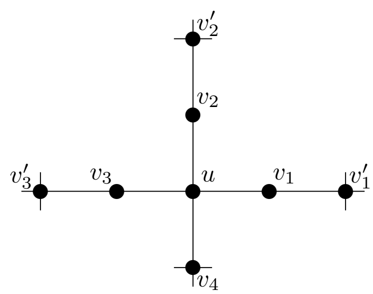


Figure 10: Configuration of Case 14

By the minimality of H there exists a linear L-colouring c of $H - \{u, v_1, v_2, v_3\}$. We extend it to u, v_1, v_2 and v_3:

– We colour u with $c(u) \in L(u)$ such that $c(u)$ is different $c(v_4)$, $c(v_1')$, $c(v_2')$, $c(v_3')$ (to avoid any bicoloured cycle) and the colours appearing twice in the neighbourhood of v_4.
– We colour v_1 with $c(v_1) \in L(v_1)$ different from $c(u)$, $c(v_1')$, $c(v_4)$, and the colours appearing twice in the neighbourhood of v_1'.
– We colour v_3 with $c(v_3) \in L(v_3)$ different from $c(u)$, $c(v_3')$, $c(v_4)$, and the colours appearing twice in the neighbourhood of v_3'.

– We colour v_2 with $c(v_2) \in L(v_2)$ different from $c(u)$, $c(v_2')$, $c(v_3)$, $c(v_1)$, and the colours appearing twice in the neighbourhood of v_2'.

There is no bicoloured cycle containing u because $c(u)$ is different from $c(v_1')$, $c(v_2')$ and $c(v_3')$. Moreover, the 2-frugality at U is assured, as 3 vertices among $\{v_1, v_2, v_3, v_4\}$ cannot share the same colour. Hence H is linearly L-colourable, a contradiction.

15. There is a configuration as depicted in Figure 11 with possibly some of the vertices in $\{v_1', v_2', v_3', v_4', v_5'\}$ being identified.

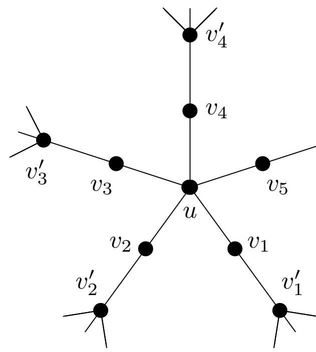


Figure 11: Configuration of Case 15

By the minimality of H there exists a linear L-colouring c of $H - \{u, v_1, v_2, v_3, v_4, v_5\}$. We extend it to u, v_1, v_2, v_3, v_4 and v_5 :

– We colour v_1 with $c(v_1) \in L(v_1)$ different from $c(v_1')$, and the colours appearing twice in the neighbourhood of v_1'.
– We colour v_2 with $c(v_2) \in L(v_2)$ different from $c(v_2')$, $c(v_1)$, and the colours appearing twice in the neighbourhood of v_2'.
– We colour v_3 with $c(v_3) \in L(v_3)$ different from $c(v_3')$, $c(v_1)$, $c(v_2)$, and the colours appearing twice in the neighbourhood of v_3'.
– We colour v_5 with $c(v_5) \in L(v_5)$ different from $c(v_5')$, $c(v_1)$, $c(v_2)$, $c(v_3)$ and the colours appearing twice in the neighbourhood of v_5'.
– We colour u with $c(u) \in L(u)$ different from $c(v_1)$, $c(v_2)$, $c(v_3)$, $c(v_5)$, and $c(v_4')$.
– We colour v_4 with $c(v_4) \in L(v_4)$ different from $c(v_4')$, $c(u)$, and the colours appearing twice in the neighbourhood of v_4'.

There is no bicoloured cycle using v_4' because $c(v_4') \neq c(u)$. Moreover there is no bicoloured cycle using both v_i and v_j, for $i < j$ and $i, j \in \{1, 2, 3, 5\}$ as $c(v_1)$, $c(v_2)$, $c(v_3)$ and $c(v_5)$ are all distionct. For the same reason, the 2-frugality at u is assured. Thus H is linearly L-colourable, a contradiction.

□

2.2 2-frugal colouring

Lemma 2. *Let H be a k-frugal-minimal graph.*

(i) If $k \geq 4$, then no 4-vertex is incident to two 2-threads.

(ii) If $k \geq \lceil \frac{\Delta}{2} \rceil + 3$, then a 4-vertex has at most one 2-neighbour.

(iii) If $k \geq \max\left(6, \lceil \frac{\Delta}{2} \rceil + 3\right)$, then a 5-vertex has at most four 2-neighbour.

Proof. Suppose that one of the assertions of Lemma 2 does not hold. Let H be a k-frugal-minimal graph for which it fails and L a k-list assignment such that H is not 2-frugally L-colourable.

(i) There is a configuration as depicted in Figure 12.

By the minimality of H there exists a 2-frugal L-colouring c of $H - \{v_1, v_2\}$. We extend it to v_1 and v_2:

– We colour v_1 with $c(v_1) \in L(v_1)$ different from $c(u), c(v_4), c(v_1')$.

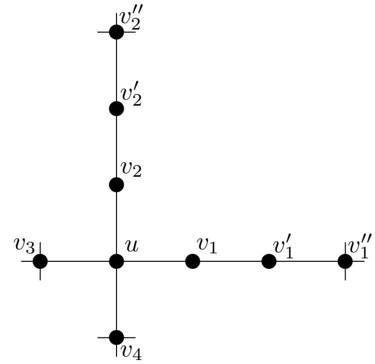


Figure 12: Configuration of Case (ii)

- We colour v_2 with $c(v_2) \in L(v_2)$ different from $c(u), c(v_3), c(v_2')$.

c is 2-frugal, given that $c(v_1) \neq c(v_4)$ and $c(v_2) \neq c(v_3)$. Hence, H is 2-frugally L-colourable, a contradiction.

(ii) There is a configuration as depicted in Figure 13 with possibly some of the vertices in $\{v_1', v_2', v_3, v_4\}$ being identified.

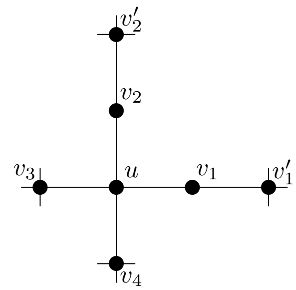


Figure 13: Configuration of Case (iii)

By the minimality of H there exists a 2-frugal L-colouring c of $H - \{v_1, v_2\}$. We extend it to v_1 and v_2:

- We colour v_1 with $c(v_1) \in L(v_1)$ different from $c(u)$, $c(v_4)$, $c(v_1')$ and the colours appearing twice in the neighbourhood of v_1'.
- We colour v_2 with $c(v_2) \in L(v_2)$ different from $c(u)$, $c(v_3)$, $c(v_2')$ and the colours appearing twice in the neighbourhood of v_2'.

c is 2-frugal, given that $c(v_1) \neq c(v_4)$ and $c(v_2) \neq c(v_3)$. Hence, H is 2-frugally L-colourable, a contradiction.

(iii) There is a configuration as depicted in Figure 14 with possibly some vertices in $\{v_1', v_2', v_3', v_4', v_5'\}$ being identified.

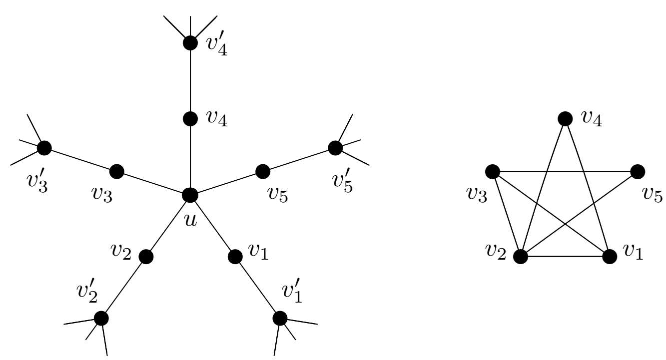


Figure 14: Configuration of Case (iv) and its auxiliary graph

By the minimality of H there exists a 2-frugal L-colouring c of $H - \{u, v_1, v_2, v_3, v_4, v_5\}$. We extend it to u, v_1, v_2, v_3, v_4 and v_5 :

- We colour v_1 with $c(v_1) \in L(v_1)$ different from $c(v_1')$, and the colours appearing twice in the neighbourhood of v_1'.

– We colour v_2 with $c(v_2) \in L(v_2)$ different from $c(v_2')$, $c(v_1)$ and the colours appearing twice in the neighbourhood of v_2'.
– We colour v_3 with $c(v_3) \in L(v_3)$ different from $c(v_3')$, $c(v_1)$, $c(v_2)$ and the colours appearing twice in the neighbourhood of v_3'.
– We colour v_4 with $c(v_4) \in L(v_4)$ different from $c(v_4')$, $c(v_1)$,$c(v_2)$ and the colours appearing twice in the neighbourhood of v_4'.
– We colour v_5 with $c(v_5) \in L(v_5)$ different from $c(v_5')$, $c(v_2)$,$c(v_3)$ and the colours appearing twice in the neighbourhood of v_5'.
– We colour u with $c(u) \in L(u)$ different from $c(v_1)$, $c(v_2)$, $c(v_3)$, $c(v_4)$ and $c(v_5)$.

The colouring is 2-frugal at u because no three vertices of $\{v_1, v_2, v_3, v_4, v_5\}$ can share the same colour as there is no stable set of size 3 in the auxiliary graph depicted on the right of Figure 14. Thus H is 2-frugally L-colourable, a contradiction.

□

3 Main results

3.1 Linear colouring – asymptotic result

In this subsection, we prove the following theorem:

Theorem 2. *Let G be a graph of maximum degree at most Δ.*

1. *If $Mad(G) < 3 - \frac{3}{\Delta+1}$ and $\Delta \geq 8$, then $\Lambda^l(G) \leq \lceil \frac{\Delta}{2} \rceil + 1$.*
2. *If $Mad(G) < 3 - \frac{9}{4\Delta+3}$ and $\Delta \geq 7$, then $\Lambda^l(G) \leq \lceil \frac{\Delta}{2} \rceil + 2$.*
3. *If $Mad(G) < 3$ and $\Delta \geq 12$ then $\Lambda^l(G) \leq \lceil \frac{\Delta}{2} \rceil + 3$.*

Proof. The proof of the three statements are similar.

We assume the existence of a counter-example G with maximum degree at most Δ such that $Mad(G) < 3 - \epsilon$ (we first consider ϵ as a variable). G then contains a subgraph H which is $(\lceil \frac{\Delta}{2} \rceil + q)$-linear-minimal with $q = 1$, 2 and 3 depending one the statement. We give to each vertex v of H an initial charge $w(v)$ equal to its degree $deg_H(v)$ in H. The average charge is then equal to the average degree of H which is at most $Mad(G)$.

We then use the following discharging rule :

- Every d-vertex, $d \geq 3$, gives $\alpha_d = \frac{d-(3-\epsilon)}{d}$ to its 2-neighbours.

We shall prove that after the discharging phase every vertex v has final charge $w^*(v)$ at least $3 - \epsilon$ for some $\epsilon \geq 0$ to be determined. This implies that

$$Ad(H) = \frac{\sum_{v \in V(H)} w(v)}{|V(H)|} = \frac{\sum_{v \in V(H)} w^*(v)}{|V(H)|} \geq 3 - \epsilon$$

which contradicts $Mad(G) < 3 - \epsilon$.

By Lemma 1-1, there is no (≤ 1)-vertex. For any $d \geq 3$, every d-vertex send at least d times α_d, so its final charge is at least $d - d \cdot \alpha_d = 3 - \epsilon$.

Let us now examine the final charge of 2-vertices. We set $d_q' = 2(\lceil \frac{\Delta}{2} \rceil + q - d) + 1$. By Lemma 1-2, every 2-vertex having a d-neighbour has also a $(\geq d_q')$-neighbour. Observe that $d_q' > \Delta$ if $d \leq q$, thus no 2-vertex has a $(\leq q)$-neighbour.

If $q \geq 2$, then a 2-vertex v has no 2-neighbour. Hence by Lemma 1-2, it has a d_1-neighbour and a d_2-neighbour, with $2 < d = d_1 \leq d_q' \leq d_2$. Since $\alpha_3 \leq \alpha_4 \leq \cdots \leq \alpha_\Delta$, the final charge of v is at least $w^*(v) = 2 + \alpha_{d_1} + \alpha_{d_2} \geq 2 + \alpha_d + \alpha_{d_q'}$.

If $q = 1$ a 2-vertex has either no 2-neighbour and as above its final charge is at least $2 + \alpha_d + \alpha_{d_q'}$ or it has a 2-neighbour and, by Lemma 1-2, its other neighbour is a Δ-vertex, so it final charge is at least $2 + \alpha_\Delta$.

Hence, to prove Theorem 2, it is sufficient to show that $2 + \alpha_d + \alpha_{d_q'} \geq 3 - \epsilon$, for all $d \leq d_q'$ and also $2 + \alpha_\Delta \geq 3 - \epsilon$ when $q = 1$.

$2 + \alpha_d + \alpha_{d_q'} \geq 3 - \epsilon$ is equivalent to $P(d) \geq 0$ with $P(d) = (1+\epsilon)dd_q' + (\epsilon - 3)(d + d_q')$. Since $d_q' = 2(\lceil \frac{\Delta}{2} \rceil + q - d) + 1$, $P(d)$ is a polynomial of degree 2 in d of the form $-2(1-\epsilon)d^2 + A \cdot d + B$ for some constant A and B (note that the coefficient of d^2 is negative). Hence to verify that $P(d) \geq 0$ for all possible values of d it suffices to prove it for the smallest and largest d such that $d \leq d_q'$, namely $\max(3, q+1)$ and $\frac{\Delta+2q+2}{3}$ (it actually is $\frac{\Delta+2q+2}{3}$ if Δ is odd and $\frac{\Delta+2q+1}{3}$ if Δ is even).

If $q \geq 2$, we obtain the following two conditions:

$$\begin{aligned}\epsilon &\geq \frac{3(q+1)-(q-2)\Delta}{q+1+(q+2)\Delta}\\ \epsilon &\geq \frac{16-2q-\Delta}{8+2q+\Delta}\end{aligned}$$

For $q = 3$ and $\Delta \geq 12$ the right hand sides of these two inequalities are negative. So they are satisfied for $\epsilon = 0$, which proves Theorem 2-3. For $q = 2$ and $\Delta \geq 7$ then $\frac{9}{3+4\Delta} \geq \frac{12-\Delta}{12+\Delta}$ and so the above inequalities are satisfied for $\epsilon = \frac{9}{3+4\Delta}$. This proves Theorem 2-2.

If $q = 1$, we have the three following conditions, the first two given by $P(d)$ and the third one by $2 + \alpha_\Delta \geq 3 - \epsilon$.

$$\begin{aligned}\epsilon &\geq \frac{9}{4\Delta-5}\\ \epsilon &\geq \frac{14-\Delta}{10+\Delta}\\ \epsilon &\geq \frac{3}{\Delta+1}\end{aligned}$$

But $\frac{3}{\Delta+1} \geq \frac{9}{4\Delta-5}$ and $\frac{3}{\Delta+1} \geq \frac{14-\Delta}{10+\Delta}$ when $\Delta \geq 8$. So the above inequalities are satisfied for $\epsilon = \frac{3}{\Delta+1}$. This proves Theorem 2-1.

□

3.2 Graphs of small maximum degree

In this subsection we prove the following theorem.

Theorem 3. *Let G be a graph with maximum degree at most Δ:*

1. *If $\Delta \geq 5$ and $Mad(G) < \frac{39}{16}$, then $\Lambda^l(G) \leq \lceil \frac{\Delta}{2} \rceil + 1$.*
2. *If $\Delta \geq 7$ and $Mad(G) < \frac{48}{19}$, then $\Lambda^l(G) \leq \lceil \frac{\Delta}{2} \rceil + 1$.*
3. *If $\Delta \geq 5$ and $Mad(G) < \frac{60}{23}$, then $\Lambda^l(G) \leq \lceil \frac{\Delta}{2} \rceil + 2$.*
4. *If $Mad(G) < \frac{14}{5}$, then $\Lambda^l(G) \leq \lceil \frac{\Delta}{2} \rceil + 3$.*
5. *If $Mad(G) < 3$. Then $\Lambda^l(G) \leq \lceil \frac{\Delta}{2} \rceil + 4$.*

Proof. The proofs of all statements are similar: we assume the existence of a counter-example G such that $Mad(G) < M$ (we first consider M as a variable), from which we deduce the existence of a subgraph H which is k-linear-minimal. We then use the discharging method to reach a contradiction.

We give to each vertex v of H an initial charge $w(v)$ equal to its degree $deg_H(v)$ in H. The average charge is then equal to the average degree of H which is at most $Mad(G)$. Then, we define discharging rules by which vertices will exchange some of their charge, keeping the average constant. We then want to prove with the help of the lemmas of the previous section that each vertex v has a final charge $w^*(v)$ at least M and so strictly greater than the average charge, which is a contradiction.

Of course, we want to find rules such that M is as large as possible in each case : for this reason, the following proofs actually define the constraints of a Linear Programme in which M is the objective value, and whose variables are the charges exchanged by the vertices during the discharging phase. At the end of each proof, we give an optimal solution of the given Linear Programme which proves the results.

1. Let G be a graph with maximum degree $\Delta \geq 5$ such that $Mad(G) < M$. Set $k_1 = \lceil \frac{\Delta}{2} \rceil + 1$. Suppose by way of contradiction that $\Lambda^l(G) > k_1$. Then G has a subgraph H which is k_1-linear-minimal.

Let us assign to every vertex of H an initial charge $w(v) = deg_H(v)$. Then $\sum_{v \in V(H)} w(v) = \sum_{v \in V(H)} d(v) = Ad(H) \cdot |V(H)|$. We now apply the following discharging rules.

Rule 1. A 2-vertex having two 3-neighbours receives α_3 from each of them.

Rule 2. A 2-vertex having only one 3-neighbour receives α'_3 from it.

Rule 3. A 2-vertex having a 2-neighbour and a (≥ 4)-neighbour receives α_4 from it.

Rule 4. A 2-vertex having a (≥ 3)-neighbour and a (≥ 4)-neighbour receives α'_4 from its (≥ 4)-neighbour.

At the end we want that the final charge of every vertex is at least M. This implies

$$Ad(H) = \frac{\sum_{v\in V(H)} w(v)}{|V(H)|} = \frac{\sum_{v\in V(H)} w^*(v)}{|V(H)|} \geq M$$

which contradicts $Mad(G) < M$.

We now define constraints on M and the α_i and α'_i guaranteeing the the final charge $w^*(v)$ of every vertex v of H is at least M.

As there are no (≤ 1)-vertices by Lemma 1-1, let us examine the final charge $w^*(v)$ of a (≥ 2)-vertex v.

- If v is a 2-vertex then by Lemma 1-3 and 1-4 it has either two 3-neighbours, or one 3-neighbour and one (≥ 4)-neighbour, or one 2-neighbour and one (≥ 4)-neighbour. In the first case $w^*(v) = 2 + 2\alpha_3$, on the second $w^*(v) \leq 2 + \alpha'_3 + \alpha'_4$ and in the last one $w^*(v) = 2 + \alpha_4$. So the required constraints are

$$M \leq 2 + 2\alpha_3 \text{ and } M \leq 2 + \alpha'_3 + \alpha'_4 \text{ and } M \leq 2 + \alpha_4.$$

- If v is a 3-vertex adjacent to three 2-neighbours, none of them can be adjacent to another 3-vertex by Lemma 1-5, and so will give at most $3\alpha'_3$. If it is adjacent to two 2-neighbours then it gives at most $\max\{2\alpha_3, 2\alpha'_3\}$. Hence, we obtain the constraints

$$M \leq 3 - 3\alpha'_3 \text{ and } M \leq 3 - 2\alpha_3.$$

- If v is a 4-vertex, by Lemma 1-6 it can not be adjacent to four 2-threads. Then it gives at most $3\alpha_4 + \alpha'_4$. (Here we assume implicitly that $\alpha'_4 \leq \alpha_4$ which is intuitively true but a priori not proved. However this inequality is satisfied by the solution giving the optimal value of M and so our assumption is a posteriori correct). Thus we obtain the constraint $M \leq 4 - 3\alpha_4 - \alpha'_4$.

- If v is a k-vertex with $k \geq 5$, it will give at most give $k\alpha_4$ yielding the constraints $M \leq k - k\alpha_4$.

The optimal value $M = \frac{39}{16}$ is obtained for $\alpha_3 = \frac{7}{32}$, $\alpha'_3 = \frac{3}{16}$, $\alpha_4 = \frac{7}{16}$ and $\alpha'_4 = \frac{1}{4}$.

2. Let G be a graph with maximum degree $\Delta \geq 7$ such that $Mad(G) < M$. Set $k_1 = \lceil \frac{\Delta}{2} \rceil + 1$. Suppose by way of contradiction that $\Lambda^l(G) > k_1$. Then G has a subgraph H which is k_1-linear-minimal.

Let us assign to every vertex of H an initial charge $w(v) = deg_H(v)$ and apply the following discharging rules.

Rule 1. A 3-vertex sends α_3 to each of its 2-neighbours.

Rule 2. A (≥ 4)-vertex sends α_4 to its 2-neighbours which are in a 2-thread and α'_4 to its other 2-neighbours.

Let us now define constraints to ensure that the final charge $w^*(v)$ of every vertex v of H is at least M, a contradiction. Again there are no (≤ 1)-vertices by Lemma 1-1.

- Suppose that v is a 2-vertex. Then by Lemma 1-3 it has at most one 2-neighbour. If v has one 2-neighbour then, by Lemma 1-4, it has a (≥ 4)-neighbour from which it receives α_4. We obtain the constraint $M \leq 2 + \alpha_4$. If v has no 2-neighbour then, by Lemma 1-9, it has a (≥ 4)-neighbour from which it receives α'_4. Its other neighbour is a (≥ 3)-neighbour from which it receives α_3. Hence we get $M \leq 2 + \alpha'_4 + \alpha_3$.

- If v is a 3-vertex it sends at most α_3 to each neighbour. This yields the constraints $M \leq 3 - 3\alpha_3$.

- If v is a 4-vertex then by Lemma 1-7, it is incident to no 2-thread, yielding $M \leq 4 - 4\alpha'_4$.

- If v is a 5-vertex then by Lemma 1-8, it is incident to at most four 2-threads, yielding $M \leq 5 - 4\alpha_4 - \alpha'_4$ (again we implicitly assume $\alpha_4 \geq \alpha'_4$ which is satisfied by the solution giving the optimal value).

- If v is a k-vertex with $k \geq 6$, it sends at most α_4 to each neighbour. Thus $M \leq k - k\alpha_4$.

The optimal value $M = \frac{48}{19}$ is obtained for $\alpha_3 = \frac{3}{19}$, $\alpha_4 = \frac{10}{19}$ and $\alpha'_4 = \frac{7}{19}$.

3. Let G be a graph with maximum degree $\Delta \geq 5$ such that $Mad(G) < M$. Set $k_2 = \lceil \frac{\Delta}{2} \rceil + 2$. Suppose by way of contradiction that $\Lambda^l(G) > k_2$. Then G has a subgraph H which is k_2-linear-minimal.

We assign to every vertex of H an initial charge $w(v) = deg_H(v)$ and apply the following discharging rules.

Rule 1. A 3-vertex sends α_3 to each of its 2-neighbours

Rule 2. A (≥ 4)-vertex sends α_4 to each of its 2-neighbours having no (≤ 3)-neighbour.

Rule 3. A (≥ 4)-vertex sends α'_4 to each of its 2-neighbours having a 3-neighbour.

Let us now define constraints to ensure that the final charge $w^*(v)$ of every vertex v of H is at least M, which is a contradiction.

- If v is a 2-vertex then it has no 2-neighbour by Lemma 1-11. If it has a 3-neighbour its other neighbour is a (≥ 4)-neighbour according to Lemma 1-9. yielding $M \leq 2 + \alpha_3 + \alpha'_4$. If v has no 3-neighbour, then we get $M \leq 2 + 2\alpha_4$.
- If v is a 3-vertex it sends at most $3\alpha_3$, yielding $M \leq 3 - 3\alpha_3$.
- If v is a 4-vertex then by Lemma 1-10, it has at most one 2-neighbour that has a 3-neighbour, yielding $M \leq 4 - \alpha'_4 - 3\alpha_4$ (with the assumption $\alpha'_4 \geq \alpha_4$).
- If v is a k-vertex with $k \geq 5$, it sends at most $k\alpha'_4$, yielding $M \leq k - k\alpha'_4$.

The optimal value $M = \frac{60}{23}$ is obtained for $\alpha_3 = \frac{3}{23}$, $\alpha_4 = \frac{7}{23}$ and $\alpha'_4 = \frac{11}{23}$.

4. Let G be a graph such that $Mad(G) < M$. Set $k_3 = \lceil \frac{\Delta}{2} \rceil + 3$. Suppose by way of contradiction that $\Lambda^l(G) > k_3$. Then $\Delta \geq 3$, as every graph with maximum degree at most 2 is linearly 3-choosable. Moreover G has a subgraph H which is k_3-linear-minimal.

Let us assign to every vertex of H an initial charge $w(v) = d_H(v)$ and apply the following discharging rule.

Rule 1. A (≥ 4)-vertex sends α_4 to each of its 2-neighbours

Let us now define constraints to ensure that the final charge $w^*(v)$ of every vertex v of H is at least M, which is a contradiction.

- If v is a 2-vertex then it has no (≤ 3)-neighbour by Lemmas 1-11 and 1-12. This gives $M \leq 2 + 2\alpha_4$.
- If v is a 3-vertex then its charge does not change, yielding $M \leq 3$.
- If v is a 4-vertex then by Lemma 1-13, it has at most three 2-neighbours. This yields $M \leq 4 - 3\alpha_4$.
- If v is a k-vertex with $k \geq 5$, it sends at most α_4 to each neighbour, yielding $M \leq k - k\alpha_4$.

The optimal value $M = \frac{14}{5}$ is obtained for $\alpha_4 = \frac{2}{5}$.

5. Let G be a graph such that $Mad(G) < M$. Set $k_4 = \lceil \frac{\Delta}{2} \rceil + 4$. Suppose by way of contradiction that $\Lambda^l(G) > k_4$. Then $\Delta \geq 3$, as every graph with maximum degree at most 2 is linearly 3-choosable. Moreover G has a subgraph H which is k_4-linear-minimal.

Let us assign to every vertex of H an initial charge $w(v) = deg_H(v)$ and apply the following discharging rule.

Rule 1. A (≥ 4)-vertex sends α_4 to each of its 2-neighbours.

Let us now define constraints to ensure that the final charge $w^*(v)$ of every vertex v of H is at least M, which is a contradiction.

- If v is a 2-vertex then it has no (≤ 3)-neighbour by Lemmas 1-11 and 1-12. This gives $M \leq 2 + 2\alpha_4$.
- If v is a 3-vertex, its charge is unchanged. This gives $M \leq 3$.
- If v is a 4-vertex then by Lemma 1-14, it has at most two 2-neighbours, yielding $M \leq 4 - 2\alpha_4$.
- If v is a 5-vertex then by Lemma 1-15, it has at most four 2-neighbours. This gives $M \leq 5 - 4\alpha_4$.
- If v is a k-vertex with $k \geq 6$, it sends at most $k\alpha_4$. Thus $M \leq k - k\alpha_4$.

The optimal value $M = 3$ is obtained for $\alpha_4 = \frac{1}{2}$.

□

3.3 2-frugal colouring

In this subsection we prove the following theorem.

Theorem 4. *Let G be a graph with maximum degree (at most) Δ*

1. *If $\Delta \geq 7$ and $Mad(G) < \frac{5}{2}$, then $\Phi_2^l(G) \leq \lceil \frac{\Delta}{2} \rceil + 1$.*
2. *If $Mad(G) < 3$, then $\Phi_2^l(G) \leq \lceil \frac{\Delta}{2} \rceil + 3$.*

Proof. 1. Let G be a graph with maximum degree $\Delta \geq 7$ such that $Mad(G) < M$. Set $k_1 = \lceil \frac{\Delta}{2} \rceil + 1 \geq 5$. Suppose by way of contradiction that $\Phi_2^l(G) > k_1$. Then G has a subgraph H which is k_1-frugal-minimal.

Let us assign to every vertex of H an initial charge $w(v) = d_H(v)$. Then $\sum_{v \in V(H)} w(v) = \sum_{v \in V(H)} d(v) = Ad(H) \cdot |V(H)|$. Let us call a 2-vertex *bad* if it has a 2-neighbour, and *good* otherwise. We now apply the following discharging rules.

Rule 1. 3-vertices give α_3 to each of their 2-neighbours.

Rule 2. (≥ 4)-vertices give α_4^g to each of their good 2-neighbours.

Rule 3. (≥ 4)-vertices give α_4^b to each of their bad 2-neighbours.

Let us now define constraints to ensure that the final charge $w^*(v)$ of every vertex v of H is at least M, which is a contradiction

- There are no 1-vertices by Lemma 1-1.
- If v is a 2-vertex then by Lemma 1-3 it does not have two 2-neighbours. In addition, by Lemma 1-4, no bad vertex can have a 3-neighbour, and by Lemma 1-9 a 2-vertex has at most one 3-neighbour. This gives the constraints $M \leq 2 + \alpha_3 + \alpha_4^g$ and $M \leq 2 + \alpha_4^b$.
- If v is a 3-vertex it sends at most α_3 to each of its neighbours, yielding $M \leq 3 - 3\alpha_3$.
- If v is a 4-vertex, then by Lemma 2-i it has at most one bad neighbour. Hence $M \leq 4 - \alpha_4^b - 3\alpha_4^g$ (with the assumption $\alpha_4^b \leq \alpha_4^g$).
- If v is a k-vertex with $k \geq 5$ it sends at most α_4^b to each of its neighbours (under the same assumption), yielding $M \leq k - 4\alpha_4^b$.

The optimal value $M = \frac{5}{2}$ is obtained for $\alpha_3 = \frac{1}{6}$, $\alpha_4^g = \frac{1}{3}$ and $\alpha_4^b = \frac{1}{2}$.

2. Let G be a graph such that $Mad(G) < M$. Set $k_3 = \lceil \frac{\Delta}{2} \rceil + 3$. Suppose by way of contradiction that $\Phi_2^l(G) > k_3$. Then G has a subgraph H which is k_1-frugal-minimal.

We assign to every vertex of H an initial charge $w(v) = deg_H(v)$ and apply the following discharging rule.

Rule 1. ≥ 4-vertices give α_4 to each of their 2-neighbours.

Let us now define constraints to ensure that the final charge $w^*(v)$ of every vertex v of H is at least M, which is a contradiction.

- There are no 1-vertices by Lemma 1-1.
- If v is a 2-vertex, by Lemma 1-11 it does not have 2-neighbours, and by Lemma 1-12 it can not have any 3-neighbour either. Hence $M \leq 2 + 2\alpha_4$.
- If v is a 3-vertex, $M \leq w^*(v) = 3$.
- If v is a 4-vertex, by Lemma 2-ii it has at most one 2-neighbour, yielding $M \leq 4 - \alpha_4$.
- If v is a 5-vertex, by Lemma 2-iii it has at most four 2-neighbours, giving $M \leq 5 - 4\alpha_4$.
- If v is a k-vertex with $k \geq 6$, then it sends at most α_4 to each of its neighbours. This yields $M \leq 6 - 6\alpha_4$.

The optimal value $M = 3$ is obtained for $\alpha_4 = \frac{1}{2}$. $\square$

References

[1] L. Esperet, M. Montassier and A. Raspaud. Linear choosability of graphs Discrete Math. 308(17): 3938-3950, 2008.

[2] B. Grünbaum, Acyclic colourings of planar graphs. *Israel J. Math.* 14:390–408, 1973.

[3] H. Hind, M. Molloy, and B. Reed. Colouring a graph frugally. *Combinatorica* 17:469–482, 1997.

[4] R. Yuster. Linear colouring of graphs. *Discrete Math.* 185:293–297, 1998.

[5] A. Raspaud and W. Wang. Linear coloring of planar graphs with large girth. *Discrete Mathematics*, 309(18):5678–5686, 2009.

[6] A. Raspaud and W. Wang, Linear colouring of planar graphs *Research report 138406* $http://www.labri.fr/perso/lepine/Rapports\_internes/RR-138406.ps.gz$, 2006.

Planar graphs with maximum degree $\Delta \geq 9$ are $(\Delta+1)$-edge-choosable. A short proof

Nathann Cohen* Frédéric Havet*

Abstract

We give a short proof of the following theorem due to Borodin [2]. Every planar graph G with maximum degree at least 9 is $(\Delta(G)+1)$-edge-choosable.

1 Introduction

All graphs considered in this paper are simple and finite. A *proper edge-colouring* of a graph G is a mapping ϕ from $E(G)$ into a set S of *colours* such that incident edges have different colours. If $|S| = k$, then f is a proper *k-edge-colouring*. A graph is *k-edge-colourable* if it has a proper k-edge-colouring. The *chromatic index* $\chi'(G)$ of a graph G is the least k such that G is k-edge-colourable.

Since edges sharing an endvertex need different colours, $\chi'(G) \geq \Delta(G)$, where $\Delta(G)$ denotes the maximum degree of G. The celebrated Vizing's Theorem [13] (also shown independently by Gupta [5]) states that $\chi'(G) \in \{\Delta(G), \Delta(G)+1\}$.

Theorem 1 (Vizing [13])**.** *If G is a graph then* $\Delta(G) \leq \chi'(G) \leq \Delta(G)+1$.

An *edge-list-assignment* of a graph G is a function L that assigns to each edge $e \in E(G)$ a list of colours $L(e)$. An edge-list-assignment is *k-uniform* if each list is of size at least k. An *L-edge-colouring* of G is a proper edge-colouring f such that $\forall v \in V(G), f(v) \in L(v)$. A graph G is *L-edge-colourable* if there exists an L-edge-colouring of G. It is *k-edge-choosable* if it is L-colourable for every k-uniform edge-list-assignment L. The *choice index* or *list chromatic index* $ch'(G)$ is the least k such that G is k-edge-choosable.

One of the most celebrated conjectures on graph colouring is the List Colouring Conjecture asserting that the chromatic index always equals the list chromatic index.

Conjecture 2 (List Colouring Conjecture)**.** *For every graph G,* $\chi'(G) = ch'(G)$.

Bollobás and Harris [1] proved that $ch'(G) < c\Delta(G)$ when $c > 11/6$ for sufficiently large Δ. Using probabilistic methods, Kahn [9] proved Conjecture 2 asymptotically: $ch'(G) \leq (1+o(1))\Delta(G)$. The error term was sharpened by Häggkvist and Janssen [7]: $ch'(G) \leq \Delta(G) + O(\Delta(G)^{2/3}\sqrt{\log \Delta(G)})$ and later by Molloy and Reed [10]: $ch'(G) \leq \Delta(G) + O(\Delta(G)^{1/2}(\log \Delta(G))^4)$. Galvin [6] proved the List Colouring Conjecture for bipartite graphs. (See also Slivnik [12]).

The List Colouring Conjecture and Vizing's Theorem imply the following conjecture :

*Projet Mascotte, I3S(CNRS, UNSA) and INRIA, 2004 route des lucioles, BP 93, 06902 Sophia-Antipolis Cedex, France. nathann.cohen@sophia.inria.fr ; fhavet@sophia.inria.fr. Partially supported by the ANR Blanc AGAPE and ANR Blanc International-Taiwan GRATEL.

Conjecture 3. *For any graph G, $ch'(G) \leq \Delta(G)+1$.*

This conjecture holds easily when $\Delta(G) \leq 2$. It is also true when $\Delta(G)=3$ as in this case the line-graph $L(G)$ of G has maximum degree 4 and so $ch'(G)=ch(L(G)) \leq 5$ as shown by Juvan et al. [8]. Borodin [2] settled this conjecture for planar graphs of maximum degree at least 9.

Theorem 4 (Borodin [2])**.** *If G is a planar graph of maximum degree at least 9, then G is $(\Delta(G)+1)$-edge-choosable.*

This theorem does not imply the List Colouring Conjecture for planar graphs of large maximum degree. Indeed, Sanders and Zhao [11] showed that a planar graph G with maximum degree at least 7 is $\Delta(G)$-edge-colourable. Vizing's Edge-Colouring Conjecture [14] asserts that $\Delta(G)$-edge-colourability also holds for planar graphs with maximum degree 6. Proving this for $\Delta(G)=6$ would be best possible as for $k \in \{2,3,4,5\}$, there are some planar graphs with maximum degree k and chromatic index equal to $k+1$ [14].

Borodin, Kostochka and Woodall [3] showed that if G is planar and $\Delta(G) \geq 12$, then $ch'(G) \leq \Delta(G)$, thus proving the List Colouring Conjecture for planar graphs with maximum degree at least 12. Another proof was given by Cole, Kowalik and Škrekovski [4]; it yields a linear-time algorithm to L-edge-colour a planar graph G for any $\max\{\Delta(G),12\}$-uniform edge-list-assignment. Conjecture 3 is still open for planar graphs with maximum degree between 5 and 8, and it is still unknown whether every planar graph with maximum degree between 6 and 11 is $\Delta(G)$-edge-choosable.

In this paper, we give a short proof of Theorem 4.

2 Proof of Theorem 4

Our proof uses the discharging method.

A vertex of degree d (respectively at least d, respectively at most d) is said to be a *d-vertex* (respectively a *d^+-vertex*, respectively a *d^--vertex*). The notion of a *d-face* (respectively a *d^+-face*, respectively a *d^--face*) is defined analogously regarding the length of a face.

Consider a minimal counterexample G to the theorem. Let L be a $(\Delta(G)+1)$-uniform ledge-list-assignment such that G is not L-edge-colourable. The graph G has no edge uv such that $d(u)+d(v) \leq \Delta(G)+2$, since otherwise any L-colouring of $G \setminus uv$ could be extended to one of G by giving to uv a colour distinct from the colours of its at most $\Delta(G)$ adjacent edges. In particular, $\delta(G) \geq 3$, and for $i \geq 3$ the neighbours of an i-vertex have degree at least $\Delta(G)+3-i$.

For each i, let V_i be the set of i-vertices.

Claim 1. $|V_{\Delta(G)}| > 2|V_3|$.

Proof. Let F the set of edges in G having one endvertex of degree 3 (hence the other endvertex of degree $\Delta(G)$). Let H be the bipartite subgraph with vertex set $V_3 \cup V_{\Delta(G)}$ and edge set F.

We show first that H is a forest. Suppose by way of contradiction that H has a cycle C. Since H is bipartite, C is even. By minimality of G, $G \setminus E(C)$ has an L-edge-colouring. Now every edge of C has at least two available colours since it is incident to $\Delta(G)+1$ edges, of which $\Delta(G)-1$ are coloured. Since even cycles are 2-edge-choosable, one can extend the L-edge-colouring to G, which is a contradiction.

Now, since every vertex of V_3 has degree 3 in H, we conclude that $|E(H)| = 3|V_3|$, and hence $|V_\Delta(G)| + |V_3| > 3|V_3|$. □

Let us assign to each vertex or face a charge equal to its degree (or length) minus 4. It follows easily from Euler's Formula that $\sum_{x\in V(G)}(d(x)-4)+\sum_{x\in F(G)}(d(x)-4)=-8$. Let us now discharge along the following rules:

(R1) Every $\Delta(G)$-vertex gives $1/2$ to a common pot from which each 3-vertex receives 1;

(R2) Every 8^+-vertex gives $1/2$ to each of its incident 3-faces;

(R3) Every d-vertex with $d\in\{5,6,7\}$ gives $\frac{d-4}{d}$ to each of its incident 3-faces.

We show that the final charge $f(x)$ for every vertex or face is nonnegative. We also show that the final charge of the common pot is nonnegative. This implies that the total final charge is nonnegative; since the total final charge equals the total initial charge, this is a contradiction.
• Since $|V_{\Delta(G)}|>2|V_3|$ by Claim 1, the charge of the common pot is positive.
• Let x be a d-vertex.
If $d=3$, then x receives 1 from the pot and gives no charge away, so $f(x)\geq 0$. If $d=4$, the charge of x does not change, so $f(x)=d-4=0$. If $d\in\{5,6,7\}$, then x sends at most $\frac{d-4}{d}$ to each of its incident face so $f(x)\geq d(1-\frac{d-4}{d})-4\geq 0$. If $8\leq d\leq\Delta(G)-1$, then x sends at most $1/2$ to each of its incident faces, so $f(x)\geq d-4-d/2\geq 0$. If $d=\Delta(G)$, then x loses charge $1/2$ to the pot and $1/2$ to each incident 3-face, so $f(x)\geq d-4-d/2-1/2\geq 0$, since $d\geq 9$.
• Let x be a d-face.
If $d\geq 4$, then its charge does not change so $f(x)=d(x)-4\geq 0$. Suppose now that $d=3$. If x contains a 4^--vertex, then the two other neighbours have degree at least $\Delta(G)-1$, so x receives $1/2$ from each of them. Thus $f(x)=3-4+2\times 1/2=0$. If x contains a 5-vertex then its two other vertices have degree at least $\Delta(G)-2$ which is at least 7. Hence x receives at least $\frac{1}{5}$ from its 5-vertex and at least $\frac{3}{7}$ from the other two vertices. So $f(x)\geq 3-4+1/5+2\times 3/7>0$. Otherwise, all the vertices incident to x are 6^+-vertices. Hence $f(x)\geq 3-4+3\times 1/3=0$.

Acknowledgement

The authors are grateful to the two anonymous referees and the editor who helped to improve the presentation of the paper.

References

[1] B. Bollobás and A. J. Harris. List colorings of graphs. *Graphs and Combin.* 1:115-127, 1985.

[2] O. V. Borodin. A generalization of Kotzig's theorem and prescribed edge coloring of planar graphs. *Mat. Zametki* 48:22-28, 1990,. (In Russian).

[3] O. V. Borodin, A. V. Kostochka, and D.R. Woodall. List edge and list total colourings of multigraphs. *J. Combin. Theory Ser. B* 71(2):184–204, 1997.

[4] R. Cole, L. Kowalik, and R. Škrekovski. A generalization of Kotzig's theorem and its application. *SIAM J. Discrete Math* 21(1):93–106, 2007.

[5] R. P. Gupta. The chromatic index and the degree of a graph. *Not. Amer. Math. Soc.* 13:719, 1966.

[6] F. Galvin. The list chromatic index of a bipartite multigraph. *J. of Combin. Theory, Ser. B* 63:153–158, 1995.

[7] R. Häggkvist and J. C. M. Janssen. New bounds on the list-chromatic index of the complete graph and other simple graph. *Combin. Probab. Comput.* 6:295–313, 1997.

[8] M. Juvan, B. Mohar, and R. Škrekovski. Graphs of degree 4 are 5-edge-choosable. *J. Graph Theory* 32:250–262, 1999.

[9] J. Kahn. Asymptotically good list colorings. *J. Combin. Theory Ser. A* 59: 31–39, 1992.

[10] M. Molloy and B. Reed. Near-optimal list colorings. *Proceedings of the Ninth International Conference "Random Structures and Algorithms" (Poznan, 1999). Random Structures Algorithms* 17(3-4):376–402, 2000.

[11] D. P. Sanders and Y. Zhao. Planar graphs of maximum degree seven are class I. *J. Combin. Theory Ser. B* 83(2):201–212, 2001.

[12] T. Slivnik. Short proof of Galvin's theorem on the list chromatic index of bipartite multigraph. *Comb. Prob. Comp.* 5:91–94, 1996.

[13] V. G. Vizing. On an estimate of the chromatic class of a p-graph. *Metody Diskret. Analyz.* 3:25–30, 1964.

[14] V. G. Vizing. Critical graphs with given chromatic index. *Metody Diskret. Analiz* 5:9–17, 1965. [In Russian]

Combinatorics, Probability and Computing (2011) 00, 000–000.
DOI: 10.1017/S0000000000000000 Printed in the United Kingdom

The α-Arboricity of Complete Uniform Hypergraphs

JEAN-CLAUDE BERMOND[1], YEOW MENG CHEE[2†], NATHANN COHEN[1‡] and XIANDE ZHANG[2§]

[1] Project MASCOTTE, INRIA/CNRS-I3S/UNSA, INRIA Sophia-Antipolis, 2004 Route de Lucioles, Sophia-Antipolis Cedex, France
(e-mail: jean-claude.bermond@sophia.inria.fr, nathann.cohen@sophia.inria.fr)

[2] Division of Mathematical Sciences, School of Physical & Mathematical Sciences, Nanyang Technological University, Singapore 637371
(e-mail: ymchee@ntu.edu.sg, xiandezhang@ntu.edu.sg)

α-Acyclicity is an important notion in database theory. The α-arboricity of a hypergraph $\mathcal{H}$ is the minimum number of α-acyclic hypergraphs that partition the edge set of $\mathcal{H}$. The α-arboricity of the complete 3-uniform hypergraph is determined completely.

1. Introduction

There is a natural bijection between database schemas and hypergraphs, where each attribute of a database schema D corresponds to a vertex in a hypergraph $\mathcal{H}$, and each relation R of attributes in D corresponds to an edge in $\mathcal{H}$. Many properties of databases have therefore been studied in the context of hypergraphs. One such property of databases is the important notion of α-acyclicity. Besides being a desirable property in the design of databases [2, 3, 8, 9, 10], many NP-hard problems concerning databases can be solved in polynomial time when restricted to instances for which the corresponding hypergraphs are α-acyclic [3, 16, 19]. Examples of such problems include determining global consistency, evaluating conjunctive queries, and computing joins or projections of joins.

† Research supported in part by the National Research Foundation of Singapore under Research Grant NRF-CRP2-2007-03 and by the Nanyang Technological University under Research Grant M58110040.
‡ Partially supported by the ANR Blanc AGAPE and ANR Blanc International-Taiwan GRATEL
§ Research supported in part by the National Research Foundation of Singapore under Research Grant NRF-CRP2-2007-03.

When faced with such computationally intractable problems on a general database schema, it is natural to decompose it into α-acyclic instances on which efficient algorithms can be applied. This has motivated some recent studies on the α-arboricity of hypergraphs, the minimum number of α-acyclic hypergraphs into which the edges of a given hypergraph can be partitioned [4, 14, 17].

In this paper, we give a general construction for partitioning complete uniform hypergraphs into α-acyclic hypergraphs based on Steiner systems, and completely determine the α-arboricity of complete 3-uniform hypergraphs.

2. Preliminaries

We assume familiarity with basic concepts and notions in graph theory.

Let n be a positive integer. The set $\{1,\dots,n\}$ is denoted $[n]$. Disjoint union of sets is denoted by $\sqcup$. We use $\sqcup$ in place of $\cup$ when we want to emphasize the disjointness of the sets involved in a union.

For X a finite set and k a nonnegative integer, the set of all k-subsets of X is denoted $\binom{X}{k}$, that is $\binom{X}{k} = \{K \subseteq X : |K| = k\}$. A *hypergraph* is a pair $\mathcal{H} = (X, \mathcal{A})$, where X is a finite set, and $\mathcal{A} \subseteq 2^X$. The elements of X are called *vertices* and the elements of $\mathcal{A}$ are called *edges*. The *order* of $\mathcal{H}$ is the number of vertices in X, and the *size* of $\mathcal{H}$ is the number of edges in $\mathcal{A}$. If $\mathcal{A} \subseteq \binom{X}{k}$, then $\mathcal{H}$ is said to be *k-uniform*. A 2-uniform hypergraph is just the usual notion of a *graph*. The *complete k-uniform hypergraph* $(X, \binom{X}{k})$ of order n is denoted $K_n^{(k)}$. A hypergraph is *empty* if it has no edges. The degree of a vertex v is the number of edges containing v.

A *Steiner system* $S(t,k,n)$ is a k-uniform hypergraph $(X,\mathcal{A})$ such that every $T \in \binom{X}{t}$ is contained in exactly one edge in $\mathcal{A}$.

A *group divisible design* k-GDD is a triple $(X,\mathcal{G},\mathcal{A})$, where $(X,\mathcal{A})$ is a k-uniform hypergraph, $\mathcal{G} = \{G_1,\dots,G_t\}$ is a partition of X into parts G_i, $i \in [t]$, called *groups*, such that every $T \in \binom{X}{2}$ not contained in a group is contained in exactly one edge in $\mathcal{A}$, and every $T \in \binom{X}{2}$ contained in a group is not contained in any edge in $\mathcal{A}$. The *type* of a k-GDD $(X,\mathcal{G},\mathcal{A})$ is the multiset $[|G_1|,\dots,|G_t|]$. The exponential notation $g_1^{t_1}\dots g_s^{t_s}$ is used to denote the multiset where element g_i has multiplicity t_i, $i \in [s]$.

We require the following result from Colbourn *et al.* [5] on the existence of 3-GDDs.

Theorem 2.1 (Colbourn, Hoffman, and Rees [5]). *Let g, t, and u be nonnegative integers. There exists a* 3-GDD *of type $g^t u^1$ if and only if the following conditions are all satisfied:*

1 *if $g > 0$ then $t \geq 3$, or $t = 2$ and $u = g$, or $t = 1$ and $u = 0$, or $t = 0$;*
2 $u \leq g(t-1)$ *or* $gt = 0$;
3 $g(t-1) + u \equiv 0 \pmod 2$ *or* $gt = 0$;
4 $gt \equiv 0 \pmod 2$ *or* $u = 0$;
5 $g^2\binom{t}{2} + gtu \equiv 0 \pmod 3$.

2.1. Graphs of Hypergraphs

Given a hypergraph $\mathcal{H}$, we may define the following graphs on $\mathcal{H}$.

Definition. Let $\mathcal{H} = (X, \mathcal{A})$ be a hypergraph. The *line graph* of $\mathcal{H}$ is the graph $L(\mathcal{H}) = (V, \mathcal{E})$, where $V = \mathcal{A}$ and $\mathcal{E} = \{\{A, B\} \subseteq \binom{V}{2} : A \cap B \neq \varnothing\}$.

Definition. Let $\mathcal{H} = (X, \mathcal{A})$ be a hypergraph. The *primal graph* of $\mathcal{H}$ is the graph $G(\mathcal{H}) = (X, \mathcal{E})$ such that $\{x, y\} \in \mathcal{E}$ if and only if $\{x, y\} \in A$ for some $A \in \mathcal{A}$.

A hypergraph $\mathcal{H}$ is *conformal* if for every clique K in $G(\mathcal{H})$, there is an edge in $\mathcal{H}$ that contains K. A hypergraph $\mathcal{H}$ is *chordal* if $G(\mathcal{H})$ is chordal, that is, every cycle of length at least four in $G(\mathcal{H})$ contains two nonconsecutive vertices that are adjacent.

2.2. Acyclic Hypergraphs

Graham [11], and independently, Yu and Ozsoyoglu [20, 21], defined an acyclicity property (which has come to be known as α-acyclicity) for hypergraphs in the context of database theory, via a transformation now known as the *GYO reduction*. Given a hypergraph $\mathcal{H} = (X, \mathcal{A})$, the GYO reduction applies the following operations repeatedly to $\mathcal{H}$ until none can be applied:

1 If a vertex $x \in X$ has degree one, then delete x from the edge containing it.
2 If $A, B \in \mathcal{A}$ are distinct edges such that $A \subseteq B$, then delete A from $\mathcal{A}$.
3 If $A \in \mathcal{A}$ is empty, that is $A = \varnothing$, then delete A from $\mathcal{A}$.

Definition. A hypergraph $\mathcal{H}$ is *α-acyclic* if GYO reduction on $\mathcal{H}$ results in an empty hypergraph.

The notion of α-acyclicity is closely related to conformality and chordality for hypergraphs. Beeri et al. [3] showed:

Theorem 2.2 (Beeri et al. [3]). *$\mathcal{H}$ is α-acyclic if and only if $\mathcal{H}$ is conformal and chordal.*

Let $\mathcal{H} = (X, \mathcal{A})$ be a hypergraph. Assign to every edge $\{A, B\}$ of $L(\mathcal{H})$ the weight $|A \cap B|$. We denote this weighted line graph of $\mathcal{H}$ by $L'(\mathcal{H})$. The maximum weight of a forest in $L'(\mathcal{H})$ is denoted $w(\mathcal{H})$. Acharya and Las Vergnas [1] introduced the hypergraph invariant

$$\mu(\mathcal{H}) = \sum_{A \in \mathcal{A}} |A| - \left| \bigcup_{A \in \mathcal{A}} A \right| - w(\mathcal{H}),$$

called the *cyclomatic number* of $\mathcal{H}$, and used it to characterize conformal and chordal hypergraphs.

Theorem 2.3 (Acharya and Las Vergnas [1]). *A hypergraph $\mathcal{H}$ satisfies $\mu(\mathcal{H}) = 0$ if and only if $\mathcal{H}$ is conformal and chordal.*

Theorem 2.2 and Theorem 2.3 immediately imply the following.

Corollary 2.4. *A hypergraph $\mathcal{H}$ is α-acyclic if and only if $\mu(\mathcal{H}) = 0$.*

Li and Wang [15] were unaware of these connections and rediscovered Corollary 2.4 recently. An easy consequence is that a maximum α-acyclic k-uniform hypergraph of order n has size $n-k+1$ [18]. Let $L_{k-1}(\mathcal{H})$ denote the spanning subgraph of $L'(\mathcal{H})$ containing only those edges of $L'(\mathcal{H})$ of weight $k-1$. We derive the following characterizations of maximum α-acyclic k-uniform hypergraphs.

Corollary 2.5. *A k-uniform hypergraph $\mathcal{H} = (X, \mathcal{A})$ of order n and size $n-k+1$ is α-acyclic if and only if $L(\mathcal{H})$ contains a spanning tree, each edge of which has weight $k-1$ (in other words, $L_{k-1}(\mathcal{H})$ is connected).*

Proof. By Corollary 2.4, we have

$$\begin{aligned} w(\mathcal{H}) &= \sum_{A\in\mathcal{A}} |A| - \left|\bigcup_{A\in\mathcal{A}} A\right| \\ &= (n-k+1)k - n \\ &= (n-k)(k-1). \end{aligned}$$

Since every edge in $L'(\mathcal{H})$ has weight at most $k-1$, and that a forest of $L'(\mathcal{H})$ contains at most $n-k$ edges (and with exactly $n-k$ edges if and only if the forest is a spanning tree), the corollary follows. □

An *α-acyclic decomposition* of a hypergraph $\mathcal{H} = (X, \mathcal{A})$ is a set of α-acyclic hypergraphs $\{(X, \mathcal{A}_i)\}_{i=1}^{c}$ such that $\mathcal{A}_1, \ldots, \mathcal{A}_c$ partition $\mathcal{A}$, that is, $\mathcal{A} = \bigsqcup_{i=1}^{c} \mathcal{A}_i$. The *size* of the α-acyclic decomposition is c.

Definition. The *α-arboricity* of a hypergraph $\mathcal{H}$, denoted $\alpha\mathrm{arb}(\mathcal{H})$, is the minimum size of an α-acyclic decomposition of $\mathcal{H}$.

3. Previous Work

Trivially, $\alpha\mathrm{arb}(K_n^{(1)}) = \alpha\mathrm{arb}(K_n^{(n)}) = 1$, since both $K_n^{(1)}$ and $K_n^{(n)}$ are α-acyclic. It is also known that $\alpha\mathrm{arb}(K_n^{(2)}) = \alpha\mathrm{arb}(K_n^{(n-1)}) = \lceil n/2 \rceil$ (see, for example, [4]). Li [14] also showed that $\alpha\mathrm{arb}(K_n^{(n-2)}) = \lceil n(n-1)/6 \rceil$. In general, Li [14] showed that

$$\left\lceil \frac{1}{k}\binom{n}{k-1} \right\rceil \le \alpha\mathrm{arb}(K_n^{(k)}) \le \frac{1}{2}\binom{n+1}{k-1}. \tag{3.1}$$

The upper and lower bounds in (3.1) differ by approximately a factor of $k/2$. Wang [17] conjectured the lower bound to be the true value of $\alpha\mathrm{arb}(K_n^{(k)})$.

Conjecture 3.1. $\alpha\mathrm{arb}(K_n^{(k)}) = \left\lceil \frac{1}{k}\binom{n}{k-1} \right\rceil$.

Recently, Chee et al. [4] showed that Conjecture 3.1 holds when $k = n-3$, so that Conjecture 3.1 is now known to hold for all n, when $k = 1, 2, n-3, n-2, n-1, n$. Chee et al. [4] also showed that Conjecture 3.1 holds whenever there exists a Steiner system $S(n-k, n-k+1, n)$, and that Conjecture 3.1 holds in an asymptotic sense when k is large enough. More precisely, the following was obtained.

Theorem 3.1 (Chee et al. [4]). *Let δ be a positive constant. Then for $k = n - O(\log^{1-\delta} n)$, we have*

$$\alpha\mathrm{arb}(K_n^{(k)}) = (1+o(1))\frac{1}{k}\binom{n}{k-1},$$

where the $o(1)$ is in n.

4. Decompositions based on Steiner Systems

First, note that the cardinality of the Steiner system $S(k-1, k, n)$ is precisely $\frac{1}{k}\binom{n}{k-1}$, i.e., when such a system exists, the lower bound given by equation 3.1. Therefore, the idea of our construction consists in using the blocks of a $S(k-1, k, n)$ as *centers* of our partitions of $K_n^{(k)}$ into α-acyclic hypergraphs. Each of these hypergraphs is based on a *center* C (in this case a block from the Steiner system) to which are added $n-3$ edges, each of which intersect the *center* on $k-1$ vertices (we name these hypergraphs *star-shaped*). The reader may find helpful to consult Fig.1 to illustrate the following proof.

Theorem 4.1. *If there exists an $S(k-1, k, n)$, then $\alpha\mathrm{arb}(K_n^{(k)}) = \frac{1}{k}\binom{n}{k-1}$.*

Proof. Let k and n be positive integers, $2 \le k \le n$. Let $(X, \mathcal{A})$ be an $S(k-1, k, n)$ and consider the set $\mathcal{A}' = \{(A, x) : A \in \mathcal{A} \text{ and } x \in X \setminus A\}$. Define a bipartite graph G with bipartition $\mathcal{A}' \sqcup \left(\binom{X}{k} \setminus \mathcal{A}\right)$ so that vertex $(A, x) \in \mathcal{A}'$ is adjacent to vertex $K \in \binom{X}{k} \setminus \mathcal{A}$ if and only if $K \subset A \cup \{x\}$. Thus, the neighborhood of vertex $(A, x) \in \mathcal{A}'$ is the set $N(A, x) = \{(A \cup \{x\}) \setminus \{u\} : u \in A\}$, and the neighborhood of vertex $K \in \binom{X}{k} \setminus \mathcal{A}$ is the set $N(K) = \{(A, x) : x \in K,\ A \in \mathcal{A}, \text{ and } K \setminus \{x\} \subset A\}$. Evidently, $|N(A, x)| = k$ for all $(A, x) \in \mathcal{A}'$. To see that $|N(K)| = k$ for all $K \in \binom{X}{k} \setminus \mathcal{A}$, note that each of the k $(k-1)$-subsets of K is contained in exactly one $A \in \mathcal{A}$, since $(X, \mathcal{A})$ is an $S(k-1, k, n)$. It follows that $|N(A, x)| = |N(K)| = k$, and G is k-regular. Hence, G has a perfect matching M.

Now, for each $A \in \mathcal{A}$, let us define the k-uniform hypergraph $\mathcal{H}_A = (X, \mathcal{B}_A)$, where $\mathcal{B}_A = \{A\} \cup \{K \in \binom{X}{k} \setminus \mathcal{A} : \{(A, x), K\} \in M \text{ for some } x \in X \setminus A\}$. It is easy to check that $\binom{X}{k} = \bigsqcup_{A \in \mathcal{A}} \mathcal{B}_A$. We claim that, in fact, the set of hypergraphs $\{\mathcal{H}_A\}_{A \in \mathcal{A}}$ is an α-acyclic decomposition of $(X, \binom{X}{k})$. To see this, note that $\mathcal{H}_A$ has order n and size $n-k+1$, and observe that each edge in $\mathcal{B}_A \setminus \{A\}$ intersects A in exactly $k-1$ vertices. Hence, $L_{k-1}(\mathcal{H}_A)$ is connected. It follows from Corollary 2.5 that $\mathcal{H}_A$ is α-acyclic. The size of the α-acyclic decomposition $\{\mathcal{H}_A\}_{A \in \mathcal{A}}$ is the size of an $S(k-1, k, n)$, which is precisely $\frac{1}{k}\binom{n}{k-1}$. $\square$

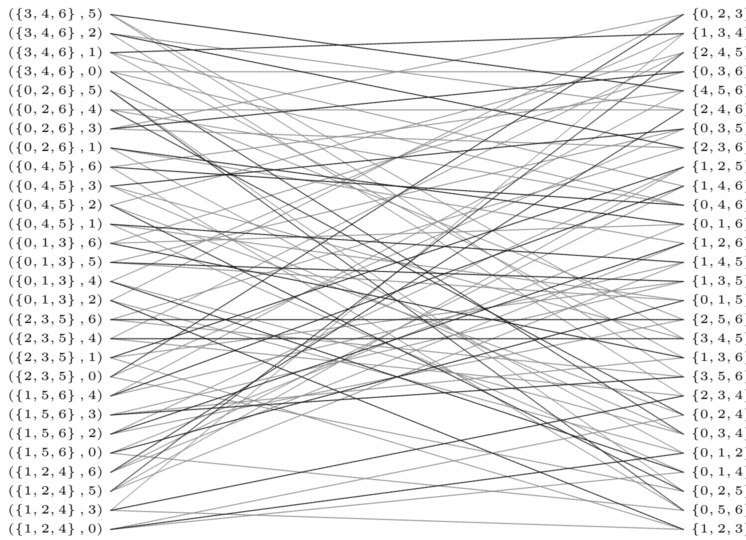


Figure 1. Case $n = 7$, $k = 3$

Corollary 4.2. *We have* $\alpha\text{arb}(K_n^{(k)}) = \frac{1}{k}\binom{n}{k-1}$ *whenever any one of the following conditions hold:*

(i) $k = 2$ *and* $n \equiv 0 \pmod 2$, *or*
(ii) $k = 3$ *and* $n \equiv 1, 3 \pmod 6$, *or*
(iii) $k = 4$ *and* $n \equiv 2, 4 \pmod 6$, *or*
(iv) $k = 5$ *and* $n \in \{11, 23, 35, 47, 71, 83, 107, 131\}$, *or*
(v) $k = 6$ *and* $n \in \{12, 24, 36, 48, 72, 84, 108, 132\}$.

Proof. For (i), note that an $S(1, 2, n)$ is a perfect matching in the complete graph K_n, and hence exists if and only if n is even. For (ii), an $S(2, 3, n)$ is a Steiner triple system and exists if and only if $n \equiv 1$ or $3 \pmod 6$ (see, for example, [7]). For (iii), an $S(3, 4, n)$ is a Steiner quadruple system, existence for which was settled by Hanani [13], who showed that $n \equiv 2$ or $4 \pmod 6$ is necessary and sufficient. For (iv)–(v), see [12, 6] for existence results. □

5. α-Arboricity of $K_n^{(3)}$

We determine $\alpha\text{arb}(K_n^{(3)})$ completely in this section. Corollary 4.2 determined $\alpha\text{arb}(K_n^{(3)})$ for all $n \equiv 1, 3 \pmod 6$, so we focus on the remaining cases of $n \equiv 0, 2, 4, 5 \pmod 6$ here.

5.1. The Case $n \equiv 0, 4 \pmod 6$

In this subsection, $n \equiv 0, 4 \pmod 6$, $n \geq 4$.

Let $X = Y \sqcup Z$, where $|Y| = n - 3$ and $Z = \{\infty_1, \infty_2, \infty_3\}$. Let $(Y, \mathcal{A})$ be an $S(2, 3, n - 3)$.

Our proof here is similar to the one given previously. Our classes, however, are now of two different kinds : not only do we need our former *star-shaped* hypergraphs whose *centers* belong to a Steiner triple system on Y, but also classes whose *centers* are two triples $\{y,\infty_1,\infty_2\}$ and $\{y,\infty_1,\infty_3\}$ (intersecting on y,∞_1), for all $y\in Y$. As previously, any edge of our α-acyclic hypergraphs intersects at least one edge of its center on exactly two vertices. The decomposition is completed by another *star-shaped* class containing the triples $\{y,\infty_2,\infty_3\}$ where $y\in X\backslash\{\infty_2,\infty_3\}$.

We define the bipartite graph Γ with bipartition $V(\Gamma)=P\sqcup Q$, where

$$P=\left(\bigcup_{A\in\mathcal{A}}\{(A,x):x\in X\setminus A\}\right)\cup \left(\bigcup_{y\in Y}\{(\{y,\infty_1,\infty_2\},\{y,\infty_1,\infty_3\},z):z\in Y\setminus\{y\}\}\right),$$

$$Q=\binom{X}{3}\setminus(\mathcal{A}\cup\{\{y,\infty_1,\infty_2\},\{y,\infty_1,\infty_3\},\{y,\infty_2,\infty_3\}:y\in Y\}\cup\{Z\}),$$

with adjacency of vertices in Γ as follows:

(i) Vertex $(\{a,b,c\},x)\in P$ is adjacent to vertices $\{a,b,x\},\{a,c,x\},\{b,c,x\}\in Q$.

(ii) Vertex $(\{y,\infty_1,\infty_2\},\{y,\infty_1,\infty_3\},z)\in P$ is adjacent to vertices $\{y,z,\infty_i\}\in Q$, $i\in[3]$.

Every vertex in P being of degree 3, let us prove the same holds for the vertices of Q. $\forall u,v\in Y$, we name A_{uv} the unique triple of $\mathcal{A}$ containing both u and v.

1 $\{a,b,c\}\subseteq Y$ is adjacent to (A_{ab},c), (A_{bc},a), and (A_{ac},b)

2 $\{a,b,\infty_i\}\in Q$ is adjacent to (A_{ab},∞_i), $(\{b,\infty_1,\infty_2\},\{b,\infty_1,\infty_2 3\},a)$, and $(\{a,\infty_1,\infty_2\},\{a,\infty_1,\infty_3\},b)$.

Hence Γ is 3-regular, and consequently has a perfect matching M.

For each $A\in\mathcal{A}$, let us define the 3-uniform hypergraph $\mathcal{H}_A=(X,\mathcal{B}_A)$, where

$$\mathcal{B}_A=\{A\}\cup\{T\in Q:\{(A,x),T\}\in M\text{ for some }x\in X\setminus A\}.$$

Then $\mathcal{H}_A$ is of order n and size $n-2$. Each edge in $\mathcal{B}_A\setminus\{A\}$ intersects A in exactly two vertices. Hence, $L_2(\mathcal{H}_A)$ is connected. It follows from Corollary 2.5 that $\mathcal{H}_A$ is α-acyclic.

In addition, for each $y\in Y$, define the 3-uniform hypergraph $\mathcal{H}_y=(X,\mathcal{B}_y)$, where

$$\mathcal{B}_y=\{\{y,\infty_1,\infty_2\},\{y,\infty_1,\infty_3\}\}\cup \{T\in Q:\ \{(\{y,\infty_1,\infty_2\},\{y,\infty_1,\infty_3\},z),T\}\in M\text{ for some }z\in Y\setminus\{y\}\}.$$

Then $\mathcal{H}_y$ is of order n and size $n-2$. In $L_2(\mathcal{H}_y)$, the vertex $\{y,\infty_1,\infty_2\}$ is adjacent to $\{y,\infty_1,\infty_3\}$, and each vertex in $\mathcal{B}_y\setminus\{\{y,\infty_1,\infty_2\},\{y,\infty_1,\infty_3\}\}$ is adjacent to vertices $\{y,\infty_1,\infty_2\}$ or $\{y,\infty_1,\infty_3\}$. Hence, $L_2(\mathcal{H}_y)$ is connected. It follows from Corollary 2.5 that $\mathcal{H}_y$ is α-acyclic.

Finally, define the 3-uniform hypergraph $\mathcal{H}=(X,\mathcal{B})$, where $\mathcal{B}=\{\{y,\infty_2,\infty_3\}:y\in Y\}\cup\{Z\}$. Note that $\mathcal{H}$ is α-acyclic, since it GYO-reduces to an empty hypergraph.

Now, we have

$$\binom{X}{3}=\left(\bigsqcup_{A\in\mathcal{A}}\mathcal{B}_A\right)\sqcup\left(\bigsqcup_{y\in Y}\mathcal{B}_y\right)\sqcup\mathcal{B},$$

so that $\{\mathcal{H}_A\}_{A\in\mathcal{A}} \cup \{\mathcal{H}_y\}_{y\in Y} \cup \{\mathcal{H}\}$ is an α-acyclic decomposition of $K_n^{(3)}$. The size of this decomposition is

$$\frac{(n-3)(n-4)}{6} + (n-3) + 1 = \frac{n(n-1)}{6},$$

which matches the lower bound in (3.1). This gives the following result.

Proposition 5.1. $\alpha\mathrm{arb}(K_n^{(3)}) = n(n-1)/6$ *for all* $n \equiv 0, 4 \pmod 6$.

5.2. The Case $n \equiv 5 \pmod 6$

In this subsection, $n \equiv 5 \pmod 6$, $n \geq 5$. Write $n = 6k+5$. Let $X = Y \sqcup \{\infty_1, \infty_2\}$, where $|Y| = 6k+3$, and let $(Y, \mathcal{G}, \mathcal{A})$ be a 3-GDD of type 3^{2k+1}, which exists by Theorem 2.1. Our construction is still based on *star-shaped* hypergraphs *centered* on the triples of the 3-GDD, but we will this time need to define *centers* consisting of 3 triples, pairwise intersecting on two elements. Also, for numerical reasons, $2k+1$ of our classes are of order only $n-2$ and size $n-4$.

In the following, only the notations G have been replaced by T

Suppose $\mathcal{G} = \{G_1, \ldots, G_{2k+1}\}$, where $G_i = \{g_{i,1}, g_{i,2}, g_{i,3}\}$, $i \in [2k+1]$. To keep our expressions succinct, we let

$$T^k_{i,j,j'} = \{g_{i,j}, g_{i,j'}, \infty_k\}$$

for $i \in [2k+1]$ $1 \leq j < j' \leq 3$ and $k \in [2]$ and

$$G_{i,j} = \{g_{i,j}, \infty_1, \infty_2\}$$

for $i \in [2k+1]$ and $j \in [3]$.

Define the bipartite graph Γ with bipartition $V(\Gamma) = P \sqcup Q$, where

$$\begin{aligned} P = {} & \left(\bigcup_{A\in\mathcal{A}} \{(A,x) : x \in X \setminus A\}\right) \cup \left(\bigcup_{G\in\mathcal{G}} \{(G,x) : x \in Y \setminus G\}\right) \cup \\ & \left(\bigcup_{i=1}^{2k+1} \{(T^1_{i,1,2}, T^1_{i,1,3}, G_{i,1}, x) : x \in Y \setminus G_i\}\right) \cup \\ & \left(\bigcup_{i=1}^{2k+1} \{(T^2_{i,1,2}, T^2_{i,2,3}, G_{i,2}, x) : x \in Y \setminus G_i\}\right), \\ Q = {} & \binom{X}{3} \setminus \Big(\mathcal{A} \cup \mathcal{G} \cup \bigcup_{\substack{i,r\\ j<j'}} \{T^r_{i,j,j'}, T^r_{i,j,j'}, T^r_{i,j,j'}\} \cup \bigcup_{i,j} G_{i,j}\Big), \end{aligned}$$

with adjacency of vertices in Γ as follows:

(i) Vertex $(\{a,b,c\}, x) \in P$ is adjacent to vertices $\{a,b,x\}, \{a,c,x\}, \{b,c,x\} \in Q$.
(ii) Vertex $(T^1_{i,1,2}, T^1_{i,1,3}, G_{i,1}, x) \in P$ is adjacent to vertices $\{g_{i,\ell}, \infty_1, x\} \in Q$, $\ell \in [3]$.
(iii) Vertex $(T^2_{i,1,2}, T^2_{i,2,3}, G_{i,2}, x) \in P$ is adjacent to vertices $\{g_{i,\ell}, \infty_2, x\} \in Q$, $\ell \in [3]$.

Every vertex in P being of degree 3, let us prove the same holds for the vertices of Q. $\forall u, v \in Y$, we name A_{uv} the unique triple of $\mathcal{A} \cup \mathcal{G}$ containing both u and v.

1 $\{a, b, c\} \subseteq Y$ is adjacent to (A_{ab}, c), (A_{bc}, a), and (A_{ac}, b).
2 $\{a, b, \infty_1\} \subseteq Q$, where $a \in G_i$ and $b \in G_{i'}$ with $i \neq i'$, is adjacent to (A_{ab}, ∞_1), $(T^1_{i,1,2}, T^1_{i,1,2}, G_{i,1}, b)$ and $(T^1_{i',1,2}, T^1_{i',1,3}, G_{i',1}, a)$
3 $\{a, b, \infty_2\} \subseteq Q$, where $a \in G_i$ and $b \in G_{i'}$ with $i \neq i'$, is adjacent to (A_{ab}, ∞_2), $(T^2_{i,1,2}, T^2_{i,2,3}, G_{i,2}, b)$ and $(T^2_{i',1,2}, T^2_{i',2,3}, G_{i',2}, a)$

Hence Γ is 3-regular, and consequently has a perfect matching M.

For each $A \in \mathcal{A}$, let us define the 3-uniform hypergraph $\mathcal{H}_A = (X, \mathcal{B}_A)$, where

$$\mathcal{B}_A = \{A\} \cup \{T \in Q : \{(A, x), T\} \in M \text{ for some } x \in X \setminus A\}.$$

Then $\mathcal{H}_A$ is of order n and size $n-2$. Each edge in $\mathcal{B}_A \setminus \{A\}$ intersects A in exactly two vertices. Hence, $L_2(\mathcal{H}_A)$ is connected. It follows from Corollary 2.5 that $\mathcal{H}_A$ is α-acyclic.

In addition, for each $G \in \mathcal{G}$, define the 3-uniform hypergraph $\mathcal{H}_G = (Y, \mathcal{B}_G)$, where $\mathcal{B}_G = \{G\} \cup \{T \in Q : \{(G, x), T\} \in M \text{ for some } x \in Y \setminus G\}$. Then $\mathcal{H}_G$ is of order $n-2$ and size $n-4$. By the same reason as for $\mathcal{H}_A$, $\mathcal{H}_G$ is α-acyclic.

Furthermore, for each $i \in [2k+1]$, define the 3-uniform hypergraphs $\mathcal{H}_i = (X, \mathcal{B}_i)$ and $\mathcal{H}'_i = (X, \mathcal{B}'_i)$, where

$$\mathcal{B}_i = \{T^1_{i,1,2}, T^1_{i,1,3}, G_{i,1}\} \cup \{T \in Q : \{(T^1_{i,1,2}, T^1_{i,1,3}, G_{i,1}, x), T\} \in M \text{ for some } x \in Y \setminus G_i\},$$
$$\mathcal{B}'_i = \{T^2_{i,1,2}, T^2_{i,2,3}, G_{i,2}\} \cup \{T \in Q : \{(T^2_{i,1,2}, T^2_{i,2,3}, G_{i,2}, x), T\} \in M \text{ for some } x \in Y \setminus G_i\}.$$

Then $\mathcal{H}_i$ and $\mathcal{H}'_i$ are each of order n and size $n-2$. In $L_2(\mathcal{H}_i)$ (respectively, $L_2(\mathcal{H}'_i)$), the vertex $T^1_{i,1,2}$ (respectively, $T^2_{i,1,2}$) is adjacent to vertices $T^1_{i,1,3}$ and $G_{i,1}$ (respectively, $T^2_{i,2,3}$ and $G_{i,2}$), and each vertex in $\mathcal{B}_i \setminus \{T^1_{i,1,2}, T^1_{i,1,3}, G_{i,1}\}$ (respectively, $\mathcal{B}'_i \setminus \{T^2_{i,1,2}, T^2_{i,2,3}, G_{i,2}\}$) is adjacent to at least one of the vertices $T^1_{i,1,2}, T^1_{i,1,3}, G_{i,1}$ (respectively, $T^2_{i,1,2}, T^2_{i,2,3}, G_{i,2}$). Hence $L_2(\mathcal{H}_i)$ (respectively, $L_2(\mathcal{H}'_i)$) is connected. It follows from Corollary 2.5 that $\mathcal{H}_i$ (respectively, $\mathcal{H}'_i$) is α-acyclic.

Finally, define the 3-uniform hypergraph $\mathcal{H} = (X, \mathcal{B})$, where

$$\mathcal{B} = \bigcup_{i=1}^{2k+1} \{T^2_{i,1,3}, T^1_{i,2,3}, G_{i,3}\}.$$

It is easy to see that $\mathcal{H}$ is α-acyclic.

Now, we have

$$\binom{X}{3} = \left(\bigsqcup_{A \in \mathcal{A}} \mathcal{B}_A \right) \sqcup \left(\bigsqcup_{G \in \mathcal{G}} \mathcal{B}_G \right) \sqcup \left(\bigsqcup_{i=1}^{2k+1} \mathcal{B}_i \right) \sqcup \left(\bigsqcup_{i=1}^{2k+1} \mathcal{B}'_i \right) \sqcup \mathcal{B},$$

so that $\{\mathcal{H}_A\}_{A \in \mathcal{A}} \cup \{\mathcal{H}_G\}_{G \in \mathcal{G}} \cup \{H_i\}_{i \in [2k+1]} \cup \{H'_i\}_{i \in [2k+1]} \cup \{\mathcal{H}\}$ is an α-acyclic decomposition of $K_n^{(3)}$. The size of this decomposition is

$$\begin{aligned} 3\binom{2k+1}{2} + (2k+1) + (2k+1) + (2k+1) + 1 &= 6k^2 + 9k + 4 \\ &= \left\lceil \frac{n(n-1)}{6} \right\rceil, \end{aligned}$$

which matches the lower bound in (3.1). This gives the following result.

Proposition 5.2. $\alpha\mathrm{arb}(K_n^{(3)}) = \lceil n(n-1)/6 \rceil$ *for all* $n \equiv 5 \pmod 6$.

5.3. The Case $n \equiv 2 \pmod 6$

We treat the remaining case of $n \equiv 2 \pmod 6$.

Lemma 5.3. $\alpha\mathrm{arb}(K_8^{(3)}) = 10$.

Proof. The lower bound in (3.1) showed that $\alpha\mathrm{arb}(K_8^{(3)}) \geq 10$. We construct an α-acyclic decomposition meeting this lower bound.

Consider the $S(2,3,7)$, $(\mathbb{Z}_7, \mathcal{A})$, with $\mathcal{A} = \{\{i, i+1, i+3\} : i \in \mathbb{Z}_7\}$. Let $\{\mathcal{H}_A\}_{A \in \mathcal{A}}$ be the α-acyclic decomposition of $(\mathbb{Z}_7, \binom{\mathbb{Z}_7}{3})$ produced by the construction of Section 4. We use this to construct an α-acyclic decomposition of $K_8^{(3)}$ as follows. Let $X = \mathbb{Z}_7 \sqcup \{\infty\}$, and let

$$\begin{aligned}
\mathcal{B}_1 &= \{\{i, i+1, \infty\} : i \in \mathbb{Z}_7 \setminus \{0\}\}, \\
\mathcal{B}_2 &= \{\{i, i+3, \infty\} : i \in \mathbb{Z}_7 \setminus \{1\}\}, \\
\mathcal{B}_3 &= \{\{i+1, i+3, \infty\} : i \in \mathbb{Z}_7 \setminus \{2\}\}, \\
\mathcal{B}_4 &= E(\mathcal{H}_{\{0,1,3\}}) \cup \{\{0, 1, \infty\}\}, \\
\mathcal{B}_5 &= E(\mathcal{H}_{\{1,2,4\}}) \cup \{\{1, 4, \infty\}\}, \\
\mathcal{B}_6 &= E(\mathcal{H}_{\{2,3,5\}}) \cup \{\{3, 5, \infty\}\}.
\end{aligned}$$

Then $\{(X, \mathcal{B}_i)\}_{i \in [6]} \cup \{\mathcal{H}_{\{3,4,6\}}, \mathcal{H}_{\{0,4,5\}}, \mathcal{H}_{\{1,5,6\}}, \mathcal{H}_{\{0,2,6\}}\}$ is an α-acyclic decomposition of $(X, \binom{X}{3})$ of size 10. $\square$

Henceforth, in what follows, let $n \equiv 2 \pmod 6$, $n \geq 14$. Write $n = 6k + 2$. Let $X = Y \sqcup \{\infty\}$, where $|Y| = 6k + 1$, and let $(Y, \mathcal{G}, \mathcal{A})$ be a 3-GDD of type $3^{2k}1^1$, which exists by Theorem 2.1. Here again, we use *star-shaped* hypergraphs *centered* on the triples of $\mathcal{A}$, but also classed whose *centers* consist of two triples intersecting in 2 elements. They will be completed with a last *star-shaped* class of order $2k + 2$ and size $2k$ (in order to reach the bound).

Suppose $\mathcal{G} = \{G_1, \ldots, G_{2k}, \{g\}\}$, where $G_i = \{g_{i,1}, g_{i,2}, g_{i,3}\}$, $i \in [2k]$. To keep our expressions succinct, we let

$$\begin{aligned}
G_i' &= \{g_{i,1}, g_{i,2}, \infty\}, \\
G_i'' &= \{g_{i,1}, g_{i,3}, \infty\}, \\
G_i''' &= \{g_{i,2}, g_{i,3}, \infty\},
\end{aligned}$$

for $i \in [2k]$. Define the bipartite graph Γ with bipartition $V(\Gamma) = P \sqcup Q$, where

$$\begin{aligned} P &= \left(\bigcup_{A \in \mathcal{A}} \{(A, x) : x \in X \setminus A\} \right) \cup \left(\bigcup_{i=1}^{2k} \{(G_i, G_i'', x) : x \in Y \setminus G_i\} \right) \cup \\ &\qquad \left(\bigcup_{i=1}^{2k} \{(G_i', G_i''', x) : x \in Y \setminus G_i\} \right) \cup \{G_i : i \in [2k]\}, \\ Q &= \binom{X}{3} \setminus (\mathcal{A} \cup \mathcal{G} \cup \{G_i', G_i'', G_i''' : i \in [2k]\}), \end{aligned}$$

with adjacency of vertices in Γ as follows:

(i) Vertex $(\{a, b, c\}, x) \in P$ is adjacent to vertices $\{a, b, x\}, \{a, c, x\}, \{b, c, x\} \in Q$.

(ii) Vertex $(G_i, G_i'', x) \in P$ is adjacent to vertices $\{g_{i,1}, g_{i,2}, x\}, \{g_{i,1}, g_{i,3}, x\}, \{g_{i,2}, g_{i,3}, x\} \in Q$.

(iii) Vertex $(G_i', G_i''', x) \in P$ is adjacent to vertices $\{g_{i,1}, \infty, x\}, \{g_{i,2}, \infty, x\}, \{g_{i,3}, \infty, x\} \in Q$.

(iv) Vertex $G_i \in P$ is adjacent to vertices $\{g_{i,j}, g, \infty\} \in Q$, $j \in [3]$.

Every vertex in P being of degree 3, let us prove the same holds for the vertices of Q. $\forall u, v \in Y$, we name A_{uv} the unique triple of $\mathcal{A}$ containing both u and v.

1 $\{a, b, c\} \subseteq Y$, where a, b, and c belong to 3 different groups, is adjacent to (A_{ab}, c), (A_{ac}, b), and (A_{bc}, a).

2 $\{a, b, c\} \subseteq Y$, where a and b belong to the same group G_i and $c \notin G_i$, is adjacent to (A_{ac}, b), (A_{bc}, a) and (G_i, G_i'', c).

3 $\{g_{i,j}, g_{i',j'}, \infty\} \in P$ (hence $i \neq i'$) is adjacent to $(A_{g_{i,j} g_{i',j'}}, \infty)$, $(G_i', G_i''', g_{i',j'})$ and $(G_{i'}', G_{i'}''', g_{i,j})$

4 $\{g_{i,j}, g, \infty\}$ is adjacent to $(A_{g_{i,j} g}, \infty)$, G_i, and (G_i', G_i''', g)

Hence Γ is 3-regular, and consequently has a perfect matching M.

For each $A \in \mathcal{A}$, let us define the 3-uniform hypergraph $\mathcal{H}_A = (X, \mathcal{B}_A)$, where

$$\mathcal{B}_A = \{A\} \cup \{T \in Q : \{(A, x), T\} \in M \text{ for some } x \in X \setminus A\}.$$

Then $\mathcal{H}_A$ is of order n and size $n - 2$. Each edge in $\mathcal{B}_A \setminus \{A\}$ intersects A in exactly two vertices. Hence, $L_2(\mathcal{H}_A)$ is connected. It follows from Corollary 2.5 that $\mathcal{H}_A$ is α-acyclic.

In addition, for each $i \in [2k]$, define the 3-uniform hypergraphs $\mathcal{H}_i = (X, \mathcal{B}_i)$ and $\mathcal{H}_i' = (X, \mathcal{B}_i')$, where

$$\begin{aligned} \mathcal{B}_i &= \{G_i, G_i''\} \cup \{T \in Q : \{(G_i, G_i'', x), T\} \in M \text{ for some } x \in Y \setminus G_i\}, \\ \mathcal{B}_i' &= \{G_i', G_i'''\} \cup \{T \in Q : \{(G_i', G_i''', x), T\} \in M \text{ for some } x \in Y \setminus G_i\}. \end{aligned}$$

Then $\mathcal{H}_i$ and $\mathcal{H}_i'$ are each of order n and size $n - 2$. By the same reason as for $\mathcal{H}_A$, $\mathcal{H}_i$ and $\mathcal{H}_i'$ are α-acyclic.

Finally, define the 3-uniform hypergraph $\mathcal{H} = (X, \mathcal{B})$, where

$$\mathcal{B} = \bigcup_{i=1}^{2k} \{T \in Q : \{G_i, T\} \in M\}.$$

It is easy to see that $\mathcal{H}$ is α-acyclic, has order $2k + 2$ and size $2k$.

Now, we have

$$\binom{X}{3} = \left(\bigsqcup_{A\in\mathcal{A}} \mathcal{B}_A\right) \sqcup \left(\bigsqcup_{i=1}^{2k} \mathcal{B}_i\right) \sqcup \left(\bigsqcup_{i=1}^{2k} \mathcal{B}'_i\right) \sqcup \mathcal{B},$$

so that $\{\mathcal{H}_A\}_{A\in\mathcal{A}} \cup \{\mathcal{H}_i\}_{i\in[2k]} \cup \{\mathcal{H}'_i\}_{i\in[2k]} \cup \{\mathcal{H}\}$ is an α-acyclic decomposition of $K_n^{(3)}$. The size of this decomposition is

$$\begin{aligned}\left(3\binom{2k}{2} + 2k\right) + 2k + 2k + 1 &= 6k^2 + 3k + 1 \\ &= \left\lceil \frac{n(n-1)}{6} \right\rceil,\end{aligned}$$

which matches the lower bound in (3.1). Together with Lemma 5.3, this gives the following result.

Proposition 5.4. $\alpha\mathrm{arb}(K_n^{(3)}) = \lceil n(n-1)/6 \rceil$ *for all* $n \equiv 2 \pmod 6$, $n \geq 8$.

5.4. Summary

Corollary 4.2(i), and Propositions 5.1, 5.2, 5.4, combine to give:

Theorem 5.5. $\alpha\mathrm{arb}(K_n^{(3)}) = \lceil n(n-1)/6 \rceil$ *for all* $n \geq 3$.

6. Conclusion

The problem of determining the α-arboricity of hypergraphs is a problem motivated by database theory. In this paper, we continue the study of the α-arboricity of complete uniform hypergraphs. We give a general construction based on Steiner systems and determine completely the value of $\alpha\mathrm{arb}(K_n^{(3)})$. Previously, $\alpha\mathrm{arb}(K_n^{(k)})$ was only known for $k = 1, 2, n-3, n-2, n-1, n$.

References

[1] B. D. Acharya and M. Las Vergnas, *Hypergraphs with cyclomatic number zero, triangulated graphs, and an inequality*, J. Combin. Theory Ser. B **33** (1982), no. 1, 52–56.

[2] C. Beeri, R. Fagin, D. Maier, A. O. Mendelzon, J. D. Ullman, and M. Yannakakis, *Properties of acyclic database schemes*, STOC 1981 – Proceedings of the 13th Annual ACM Symposium on Theory of Computing, ACM Press, 1981, pp. 355–362.

[3] C. Beeri, R. Fagin, D. Maier, and M. Yannakakis, *On the desirability of acyclic database schemes*, J. Assoc. Comput. Mach. **30** (1983), no. 3, 479–513.

[4] Y. M. Chee, L. Ji, A. Lim, B. C. Ooi, and A. K. H. Tung, *Arboricity: an acyclic hypergraph decomposition problem motivated by database theory*, Preprint, 2009.

[5] C. J. Colbourn, D. G. Hoffman, and R. Rees, *A new class of group divisible designs with block size three*, J. Combin. Theory Ser. A **59** (1992), no. 1, 73–89.

[6] C. J. Colbourn and R. Mathon, *Steiner systems*, The CRC Handbook of Combinatorial Designs (C. J. Colbourn and J. H. Dinitz, eds.), CRC Press, Boca Raton, 2nd ed., 2007, pp. 102–110.

[7] C. J. Colbourn and A. Rosa, *Triple Systems*, Oxford Mathematical Monographs, Oxford University Press, New York, 1999.

[8] R. Fagin, *Degrees of acyclicity for hypergraphs and relational database schemes*, J. Assoc. Comput. Mach. **30** (1983), no. 3, 514–550.

[9] R. Fagin, A. O. Mendelzon, and J. D. Ullman, *A simplified universal relation assumption and its properties*, ACM Trans. Database Syst. **7** (1982), no. 3, 343–360.

[10] N. Goodman and O. Shmueli, *Characterizations of tree database schemas*, Tech. report, Harvard University, Cambridge, Massachusetts, 1981.

[11] M. H. Graham, *On the universal relation*, Tech. report, University of Toronto, Toronto, Ontario, Canada, September 1979.

[12] M. J. Grannell, T. S. Griggs, and R. A. Mathon, *Steiner systems $S(5,6,v)$ with $v = 72$ and* 84, Math. Comp. **67** (1998), no. 221, 357–359, S1–S9. MR MR1451323 (98c:05020)

[13] H. Hanani, *On quadruple systems*, Canad. J. Math. **12** (1960), 145–157.

[14] H. Li, *On acyclic and cyclic hypergraphs*, Ph.D. thesis, Institute of Applied Mathematics, Chinese Academy of Science, Beijing, China, 1999.

[15] H. Li and J. Wang, *On acyclic and cyclic hypergraphs*, J. Syst. Sci. Complex. **15** (2002), no. 4, 353–362.

[16] R. E. Tarjan and M. Yannakakis, *Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs and selectively reduce acyclic hypergraphs*, SIAM J. Comput. **13** (1984), 566–579.

[17] J. Wang, *The information hypergraph theory*, Science Press, Beijing, 2008.

[18] J. Wang and H. Li, *Enumeration of maximum acyclic hypergraphs*, Acta Math. Appl. Sin. **18** (2002), no. 2, 215–218.

[19] M. Yannakakis, *Algorithms for acyclic database schemes*, VLDB 1981 – Proceedings of the 7th International Conference on Very Large Data Bases, ACM Press, 1981, pp. 82–94.

[20] C. T. Yu and M. Z. Ozsoyoglu, *An algorithm for tree-query membership of a distributed query*, COMPSAC '79: Proceedings of the IEEE Computer Software and Applications Conference, IEEE Computer Society, 1979, pp. 306–312.

[21] ———, *On determining tree-query membership of a distributed query*, Tech. Report 80-1, University of Alberta, Edmonton, Canada, January 1980.

Algorithm for Finding k-Vertex Out-trees and its Application to k-Internal Out-branching Problem*

Nathann Cohen† Fedor V. Fomin‡ Gregory Gutin§ Eun Jung Kim§
Saket Saurabh‡ Anders Yeo§

Abstract

An out-tree T is an oriented tree with only one vertex of in-degree zero. A vertex x of T is internal if its out-degree is positive. We design randomized and deterministic algorithms for deciding whether an input digraph contains a given out-tree with k vertices. The algorithms are of running time $O^*(5.704^k)$ and $O^*(6.14^k)$, respectively. We apply the deterministic algorithm to obtain a deterministic algorithm of runtime $O^*(c^k)$, where c is a constant, for deciding whether an input digraph contains a spanning out-tree with at least k internal vertices. This answers in affirmative a question of Gutin, Razgon and Kim (Proc. AAIM'08).

1 Introduction

An *out-tree* is an oriented tree with only one vertex of in-degree zero called the *root*. The k-Out-Tree problem is the problem of deciding for a given parameter k, whether an input digraph contains a given out-tree with $k \geq 2$ vertices. In their seminal work on Color Coding Alon, Yuster, and Zwick [1] provided fixed-parameter tractable (FPT) randomized and deterministic algorithms for k-Out-Tree. While Alon, Yuster, and Zwick [1] only stated that their algorithms are of runtime $O(2^{O(k)}n)$, however, it is easy to see (see Subsection 2.1), that their randomized and deterministic algorithms are of complexity[1] $O^*((4e)^k)$ and $O^*(c^k)$, where $c \geq 4e$.

The main results of [1], however, were a new algorithmic approach called Color Coding and a randomized $O^*((2e)^k)$ algorithm for deciding whether a digraph contains a path with k vertices (the k-Path problem). Chen et al. [4] proposed another approach, a randomized divide-and-conquer technique; the new appraoch allowed them to design a randomized $O^*(4^k)$-time algorithm for k-Path. To divide the technique of Chen et al. [4] uses two colors. The colors are 'symmetric', i.e., both colors play similar role and the probability of coloring each vertex in one of the colors is 0.5. In this paper, we further develop the technique of [4] by making it asymmetric, i.e., the two colors play different roles and the probability of coloring each vertex in one of the colors depends on the color. As a result, we refine the

*An extended abstract of this paper has appeared in the proceedings of COCOON'09 [6].

†INRIA – Projet MASCOTTE, 2004 route des Lucioles, BP 93 F-06902, Sophia Antipolis Cedex, France, nathann.cohen@sophia.inria.fr

‡Department of Informatics, University of Bergen, POB 7803, 5020 Bergen, Norway, fedor.fomin|saket.saurabh@ii.uib.no

§Department of Computer Science, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK, gutin|eunjung|anders@cs.rhul.ac.uk

[1] In this paper we often use the notation $O^*(f(k))$ instead of $f(k)(kn)^{O(1)}$, i.e., O^* hides not only constants, but also polynomial coefficients.

result of Alon, Yuster, and Zwick by obtaining randomized and deterministic algorithms for k-Out-Tree, of runtime $O^*(5.704^k)$ and $O^*(6.14^k)$, respectively.

It is worth to mention here two recent related results on k-Path due to Koutis [10] and Williams [16] based on an algebraic approach. Koutis [10] obtained a randomized $O^*(2^{3k/2})$-time algorithm for k-Path and Williams [16] extended his ideas resulting in a randomized $O^*(2^k)$-time algorithm for k-Path. While the randomized algorithms based on Color Coding and Divide-and-Color are not difficult to derandomize, it is not the case for the algorithms of Koutis [10] and Williams [16]. Thus, it is unknown whether there are deterministic algorithms for k-Path of runtime $O^*(2^{3k/2})$. Moreover, it is not clear whether the randomized algorithms of Koutis [10] and Williams [16] can be extended to solve k-Out-Tree.

While we believe that the study of fast algorithms for k-Out-Tree is a problem interesting on its own, we provide an application of our deterministic algorithm. The vertices of an out-tree T of out-degree zero (nonzero) are *leaves* (*internal vertices*) of T. An *out-branching* of a digraph D is a spanning subgraph of D which is an out-tree. The Minimum Leaf problem is to find an out-branching with the minimum number of leaves in a given digraph D. This problem is of interest in database systems [7] and the Hamilton path problem is its special case. Thus, in particular, Minimum Leaf is NP-hard. In this paper we will study the following parameterized version of Minimum Leaf: given a digraph D and a parameter k, decide whether D has an out-branching with at least k internal vertices. This problem denoted k-Int-Out-Branching was studied for symmetric digraphs (i.e., undirected graphs) by Prieto and Sloper [14, 15] and for all digraphs by Gutin et al. [9]. Gutin et al. [9] obtained an algorithm of runtime $O^*(2^{O(k \log k)})$ for k-Int-Out-Branching and asked whether the problem admits an algorithm of runtime $O^*(2^{O(k)})$. Note that no such algorithm has been known even for the case of symmetric digraphs [14, 15]. In this paper, we obtain an $O^*(2^{O(k)})$-time algorithm for k-Int-Out-Branching using our deterministic algorithm for k-Out-Tree and an out-tree generation algorithm.

For a set X of vertices of a subgraph H of a digraph D, $N^+_H(X)$ and $N^-_H(X)$ denote the sets of out-neighbors and in-neighbors of vertices of X in H, respectively. Sometimes, when a set has a single element, we will not distinguish between the set and its element. In particular, when H is an out-tree and x is a vertex of H which is not its root, the unique in-neighbor of x is denoted by $N^-_H(x)$. For an out-tree T, $\mathrm{Leaf}(T)$ denotes the set of leaves in T and $\mathrm{Int}(T) = V(T) - \mathrm{Leaf}(T)$ stands for the set of internal vertices of T.

2 Algorithms for k-Out-Tree

In Subsection 2.1, we introduce and analyze the randomized algorithm for k-Out-Tree by Alon, Yuster and Zwick [1]. In Subsection 2.2, we introduce and analyze a new randomized algorithm for k-Out-Tree. We derandomize our algorithm in Subsection 2.3.

2.1 Algorithm of Alon, Yuster and Zwick

Let $c : V(D) \to \{1, \dots, k\}$ be a vertex k-coloring of a digraph D and let T be a k-vertex out-tree contained in D (as a subgraph). Then $V(T)$ and T are *colorful* if no pair of vertices of T are of the same color.

The following algorithm of [1] verifies whether D contains a colorful out-tree H such that H is isomorphic to T, when a coloring $c : V(D) \to \{1, \dots, k\}$ is given. Note that a k-vertex subgraph H will be colorful with a probability of at least $k!/k^k > e^{-k}$. Thus, we can find a copy of T in D in e^k expected iterations of the following algorithm.

Algorithm 1 $\mathcal{L}(T, r)$

Require: A digraph D with a given coloring $c : V(D) \to \{1, \ldots, k\}$, an out-tree T on k vertices, a specified vertex r of D

Ensure: $\mathcal{C}_T(u)$ for each vertex u of D, which is a family of all color sets that appear on colorful copies of T in D, where u plays the role of r

1: **if** $|V(T)| = 1$ **then**
2:     **for all** $u \in V(D)$ **do**
3:         Insert $\{c(u)\}$ into $\mathcal{C}_T(u)$.
4:     **end for**
5:     Return $\mathcal{C}_T(u)$ for each vertex u of D.
6: **else**
7:     Choose an arc $(r', r'') \in A(T)$.
8:     Let T' and T'' be the subtrees of T obtained by deleting (r', r''), where T' and T'' contains r' and r'', respectively.
9:     Call $\mathcal{L}(T', r')$.
10:     Call $\mathcal{L}(T'', r'')$.
11:     **for all** $u \in V(D)$ **do**
12:         Compose the family of color sets $\mathcal{C}_T(u)$ as follows:
13:         **for all** $(u, v) \in A(D)$ **do**
14:             **for all** $C' \in \mathcal{C}_{T'}(u)$ and $C'' \in \mathcal{C}_{T''}(v)$ **do**
15:                 $C := C' \cup C''$ if $C' \cap C'' = \emptyset$
16:                 Insert C into $\mathcal{C}_T(u)$.
17:             **end for**
18:         **end for**
19:     **end for**
20:     Return $\mathcal{C}_T(u)$ for each vertex u of D.
21: **end if**

Theorem 2.1. *Let T be an out-tree on k vertices and let $D = (V, A)$ be a digraph. A subgraph of D isomorphic to T, if one exists, can be found in $O(k(4e)^k \cdot |A|)$ expected time by running the algorithm $\mathcal{L}(T, r)$ for a random coloring c iteratively.*

Proof. Let $c : V(D) \to \{1, \ldots, k\}$ be a given coloring of D and suppose T' and T'' are the subtrees of T obtained in line 8. Let $|V(T')| = k'$ and $|V(T'')| = k''$, where $k' + k'' = k$. Then $|\mathcal{C}_{T'}(u)| = \binom{k-1}{k'-1}$ and $|\mathcal{C}_{T'}(u)| = \binom{k-1}{k''-1}$. Checking $C' \cap C'' = \emptyset$ takes $O(k)$ time. Hence, lines 11-19 require at most $\binom{k}{k/2}^2 \cdot k|A| \leq k2^{2k}|A|$ operations.

Let $T(k)$ be the number of operations for $\mathcal{L}(T, r)$. We have the following recursion.

$$T(k) \leq T(k') + T(k'') + k2^{2k-2}|A| \tag{1}$$

By induction, it is not difficult to check that $T(k) \leq k4^k|A|$. Since the expected number of iterations of the algorithm $\mathcal{L}(T, r)$ is at most e^k, we achieve the claimed running time. $\square$

Let $\mathcal{C}$ be a family of vertex k-colorings of a digraph D. We call $\mathcal{C}$ an *(n, k)-family of perfect hashing functions* if for each $X \subseteq V(D)$, $|X| = k$, there is a coloring $c \in \mathcal{C}$ such that X is colorful with respect to c. One can derandomize the above algorithm of Alon et al. by using any (n, k)-family of perfect hashing functions in the obvious way. The time complexity of the derandomized algorithm depends of the size of the (n, k)-family of perfect hashing functions. Let $\tau(n, k)$ denote the minimum size of an (n, k)-family of perfect hashing functions. Nilli [12] proved that $\tau(n,k) \geq \Omega(e^k \log n/\sqrt{k})$. It is unclear whether there is an (n,k)-family of perfect hashing functions of size $O^*(e^k)$ [4], but even if it does exist, the running time of the derandomized algorithm would be $O^*((4e)^k)$.

2.2 New Algorithm for k-Out-Tree

Before we introduce our new randomized algorithm for k-Out-Tree, we will give a brief account of the basic idea behind it. Let T be an out-tree on k vertices and let D be a digraph in which we want to find a copy of T. As in the randomized algorithm by Alon, Yuster and Zwick in [1], we break T into two subtrees T_w and T_b. However, unlike the former which deletes an arc of T, we break it by choosing a "splitting vertex" denoted as v^* and furthermore the resulting two subtrees overlap exactly on this splitting vertex v^*. Next we randomly partition the digraph D into two vertex-disjoint parts D_w and D_b, and then find a copy of T_w in D_w and a copy of T_b in D_b, if one exists. If we try sufficiently many partitions of D, it is possible to find a copy of T whenever D contains one as a subgraph (with some good probability in a randomized version of the algorithm, which can be derandomized consequently).

The trouble is that the fact D_w and D_b that contain copies of T_w and T_b, respectively does not necessarily means that D contains a copy of T as a whole. We need to ensure that there exist copies of T_w and T_b that actually overlap (and overlap only) on a vertex of D corresponding to the splitting vertex v^*. To this end, we allow some vertices of D_w, say S, to be shared by D_b by considering $D_b + S$ instead of D_b. Here S is the set of vertices in D_w that could correspond to the splitting vertex v^* of T_w. When we search for a copy of T_b in $D_b + S$, only those tress isomorphic to T_b in $D_b + S$ are considered legitimate where the vertex corresponding to v^* lies in S. In other words, we convey the information S obtained in the phase for T_w-D_w to the next phase for T_b-D_b so that we do not only ensure the global connectivity of $T_w + T_b = T$ in D but also reduce the search space for finding a copy of T_b in D_b.

Moreover, by conveying the information for v^* we can save the extra effort for "merging" the solutions (i.e. copies of T_w and T_b). Rather, once we obtain a copy of T_b in $D_b + S$, it follows immediately that we have a copy of T in D. Since the number of partitions of D we need to try is a function of k, the time complexity of finding a copy of T in D can be written as $T(k,n) = f(k)(T(k',n) + T(k-k',n) + p_1(n)) + p_2(n)$, $T(1,n) = p_3(n)$, where $p_i(n)$ is polynomial in n for $i = 1,2,3$. This is why the running time of our algorithm remains polynomial in n. Making this approach efficient depends crucially on two aspects:

1. to obtain k' in the above formula as close to half of k as possible; and
2. to replace $f(k)$ with as small growing function as possible.

For the latter, we use a simple *unbalanced-partition-strategy* which will be explained later. We achieve the former goal by choosing an appropriate splitting vertex v^* and then using it to obtain a T_w-T_b split. The splitting procedure is one of the key part of our algorithm and next we describe this procedure in details.

The following lemma is well known and will be used as a basic scheme of choosing v^*.

Lemma 2.2 ([5])**.** *Let T be an undirected tree and let $w : V \to \mathbb{R}^+ \cup \{0\}$ be a weight function on its vertices. There exists a vertex $v \in V(T)$ such that the weight of every subtree T' of $T - v$ is at most $w(T)/2$, where $w(T) = \sum_{v \in V(T)} w(v)$.*

Consider a partition $n = n_1 + \cdots + n_q$, where n and all n_i are nonnegative integers and a bipartition (A, B) of the set $\{1, \ldots, q\}$. Let $d(A,B) := \left| \sum_{i \in A} n_i - \sum_{i \in B} n_i \right|$. Given a set

$Q = \{1, \ldots, q\}$ with a nonnegative integer weight n_i for each element $i \in Q$, we say that a bipartition (A, B) of Q is *greedily optimal* if $d(A, B)$ does not decrease by moving an element of one partite set into another. The following procedure describes how to obtain a greedily optimal bipartition in time $O(q \log q)$. For simplicity we write $\sum_{i \in A} n_i$ as $n(A)$.

Algorithm 2 Bipartition$(Q, \{n_i : i \in Q\})$

Require: A set $Q = \{1, \ldots, q\}$ with a nonnegative integer weight n_i, $\forall i \in Q$
Ensure: A greedily optimal bipartition (A, B) of Q

```
1: Let A := ∅, B := Q.
2: while n(A) < n(B) and there is an element i ∈ B with 0 < n_i < d(A,B) do
3:     Choose such an element i ∈ B with a largest n_i.
4:     A := A ∪ {i} and B := B − {i}.
5: end while
6: Return (A, B).
```

Lemma 2.3. *Let Q be a set of size q with a nonnegative integer weight n_i for each $i \in Q$. The algorithm* **Bipartition**$(Q, \{n_i : i \in Q\})$ *finds a greedily optimal bipartition $A \cup B = Q$ in time $O(q \log q)$.*

Proof. First we want to show that the values n_i chosen in line 3 of the algorithm do not increase during the performance of the algorithm. The values of n_i do not increase because the values of the difference $d(A, B)$ do not increase during the performance of the algorithm. In fact, $d(A, B)$ strictly decreases. To see this, suppose that the element i is selected in the present step. If $n(A \cup \{i\}) < n(B - \{i\})$, then obviously the difference $d(A, B)$ strictly decreases. Else if $n(A \cup \{i\}) > n(B - \{i\})$, we have $d(A \cup \{i\}, B - \{i\}) < n_i < d(A, B)$.

To see that the algorithm returns a greedily optimal bipartition (A, B), it is enough to observe that for the final bipartition (A, B), moving any element of A or B does not decrease $d(A, B)$. Suppose that the last movement of the element i_0 makes $n(A) > n(B)$. Then a simple computation implies that $d(A, B) < n_{i_0}$. Since the values of n_i in line 3 of the algorithm do not increase during the performance of the algorithm, $n_j \geq n_{i_0} > d(A, B)$ for every $j \in A$, the movement of any element in A would not decrease $d(A, B)$. On the other hand suppose that $n(A) < n(B)$. By the definition of the algorithm, for every $j \in B$ with a positive weight we have $n_j \geq d(A, B)$ and thus the movement of any element in B would not decrease $d(A, B)$. Hence the current bipartition (A, B) is greedily optimal.

Now let us consider the running time of the algorithm. Sorting the elements in nondecreasing order of their weights will take $O(q \log q)$ time. Moreover, once an element is moved from one partite set to another, it will not be moved again and we move at most q elements without duplication during the algorithm. This gives us the running time of $O(q \log q)$. $\square$

Now we describe a new randomized algorithm for k-Out-Tree. Let D be a digraph and let T be an out-tree on k vertices. Let us specify a vertex $t \in V(T)$ and a vertex $w \in V(D)$. We call a copy of T in D a *T-isomorphic* tree. We say that a T-isomorphic tree T_D in D is a (t, w)-tree if $w \in V(T_D)$ plays the role of t.

We first give an intuitive explanation of our algorithm before giving a formal description. To find the desired tree in the given input digraph, we first split the tree in two parts with one common vertex such that the both parts are "almost balanced" Then we randomly partition the vertices of the D in two parts with probability of a vertex lying in one part or the other depends on the sizes of the trees we obtained in the first step by splitting it on a vertex. This allows us to more or less independently look for the different parts of the tree in different parts of the partition. We finally merge them cleverly to obtain our solution.

In the following algorithm **find-tree**, we have several arguments other than the natural arguments T and D. Our next argument is a vertex t of T. The argument t indicates that we want to return, at the end of the current procedure, the set of vertices X_t such that there is a (t, w)-tree for every $w \in X_t$. The fact that $X_t \neq \emptyset$ means two points: we have a T-isomorphic tree in D, and the information X_t we have can be used to construct a larger tree which uses the current T-isomorphic tree as a building block. Here, X_t is a kind of 'joint'.

The basic strategy is as follows. We choose a pair T_A and T_B of subtrees of T such that $V(T_A) \cup V(T_B) = V(T)$ and T_A and T_B share only one vertex, namely v^*, the splitting vertex. We call recursively two '**find-tree**' procedures on subsets of $V(D)$ to ensure that the subtrees playing the role of T_A and T_B do not overlap. The first call (line 15) tries to find X_{v^*} and the second one (line 18), using the information X_{v^*} delivered by the first call, tries to find X_t.

We also need another argument to our algorithm **find-tree** which is useful while merging and that is:

- a pair consisting of $L \subseteq V(T)$ and $\{X_u : u \in L\}$, where $X_u \subset V(D)$ and X_u's are pairwise disjoint.

The arguments $L \subseteq V(T)$ and $\{X_u : u \in L\}$ form a set of information needed to argue the correctness of the algorithm. Essentially L is a set of vertices of the tree T which has been used as a splitting vertex at some point during the execution of our recursive procedure. Let T_D be a T-isomorphic tree; if for every $u \in L$, T_D is a (u, w)-tree for some $w \in X_u$ and $V(T_D) \cap X_u = \{w\}$, we say that T_D *meets the restrictions on* L. The algorithm **find-tree** intends to find the set X_t of vertices such that for every $w \in X_t$, there is a (t, w)-tree which meets the restrictions on L.

Deleting a splitting vertex v^* may produce several subtrees, and there might be many ways to divide them into two groups, namely (T_A, T_B). To make the algorithm more efficient, we try to obtain as 'balanced' a partition (T_A, T_B) as possible. The algorithm **tree-Bipartition** is used to produce a pretty 'balanced' bipartition of the subtrees. Moreover we introduce another argument to have a better complexity behavior. The argument v is a vertex which indicates whether there is a predetermined splitting vertex. If $v = \emptyset$, we do not have a predetermined splitting vertex so we find one in the current procedure. Otherwise, we use the vertex v as a splitting vertex.

Let r be the root of T. To decide whether D contains a copy of T, it suffices to run **find-tree**$(T, D, \emptyset, r, \emptyset, \emptyset)$.

Lemma 2.4. *During the performance of find-tree($T, D, \emptyset, r, \emptyset, \emptyset$), the sets X_u, $u \in L$ are pairwise disjoint.*

Proof. We prove the claim inductively. For the initial call, trivially the sets X_u, $u \in L$ are pairwise disjoint since $L = \emptyset$. Suppose that for a call find-tree$(T, D, v, t, L, \{X_u : u \in L\})$ the sets X_v, $v \in L$ are pairwise disjoint. For the first subsequent call in line 15, the sets are obviously pairwise disjoint. Consider the second subsequent call in line 18. If $v^* \in L$ before line 17, the claim is true since we convey the argument $t := v^*$ to the first subsequent call in line 15 and thus S is contained in X_{v^*}. Otherwise, observe that $X_u \subseteq V_b$ for all $u \in L \cap U_b$ and they are pairwise disjoint. Since $X_{v^*} \cap V_b = \emptyset$, the sets X_u for all $u \in L \cap U_b$ together with X_{v^*} are pairwise disjoint. $\square$

Algorithm 3 find-tree$(T, D, v, t, L, \{X_u : u \in L\})$

Require: An out-tree T on k vertices, a digraph D, $v \in \{\emptyset\} \cup V(T)$, a specified vertex $t \in V(T)$, a subset of vertices $L \subseteq V(T)$, a family of pairwise disjoint subsets $X_u \subseteq V(D)$ for each $u \in L$.

Ensure: A set of vertices $X_t \subseteq V(D)$ such that there is a (t, w)-tree for every $w \in X_t$.

if $|V(T) \setminus L| \geq 2$ **then**
    for all $u \in V(T)$**:** Set $w(u) := 0$ if $u \in L$, $w(u) := 1$ otherwise.
    if $v = \emptyset$ **then** Find $v^* \in V(T)$ such that the weight of every subtree T' of $T - v^*$ is at most $w(T)/2$ (see Lemma 2.2) **else** $v^* := v$
    (WH, BL):=tree-Bipartition(T, t, v^*, L).
    $U_w := \bigcup_{i \in WH} V(T_i) \cup \{v^*\}$, $U_b := \bigcup_{i \in BL} V(T_i)$.
    for all $u \in L \cap U_w$**:** color all vertices of X_u in white.
    for all $u \in L \cap (U_b \setminus \{v^*\})$**:** color all vertices of X_u in black.
    $\alpha := \min\{w(U_w)/w(T), w(U_b)/w(T)\}$.
    if $\alpha^2 - 3\alpha + 1 \leq 0$ (i.e., $\alpha \geq (3 - \sqrt{5})/2$, see (2) and the definition of α^* afterwards) **then** $v_w := v_b := \emptyset$
    else if $w(U_w) < w(U_b)$ **then** $v_w := \emptyset$, $v_b := v^*$ **else** $v_w := v^*$, $v_b := \emptyset$.
    $X_t := \emptyset$.
    for $i = 1$ to $\left\lceil \frac{2.51}{\alpha^{\alpha k}(1-\alpha)^{(1-\alpha)k}} \right\rceil$ **do**
        Color the vertices of $V(D) - \bigcup_{u \in L} X_u$ in white or black such that for each vertex the probability to be colored in white is α if $w(U_w) \leq w(U_b)$, and $1 - \alpha$ otherwise.
        Let V_w (V_b) be the set of vertices of D colored in white (black).
        S :=find-tree$(T[U_w], D[V_w], v_w, v^*, L \cap U_w, \{X_u : u \in L \cap U_w\})$
        if $S \neq \emptyset$ **then**
            $X_{v^*} := S$, $L := L \cup \{v^*\}$.
            S' :=find-tree$(T[U_b \cup \{v^*\}], D[V_b \cup S], v_b, t, (L \cap U_b), \{X_u : u \in (L \cap U_b)\})$.
            $X_t := X_t \cup S'$.
        end if
    end for
    Return X_t.
else $\{|V(T) \setminus L| \leq 1\}$
    if $\{z\} = V(T) \setminus L$ **then** $X_z := V(D) - \bigcup_{u \in L} X_u$, $L := L \cup \{z\}$.
    $L^o := \{$all leaf vertices of $T\}$.
    while $L^o \neq L$ **do**
        Choose a vertex $z \in L \setminus L^o$ s.t. $N_T^+(z) \subseteq L^o$.
        $X_z := X_z \cap \bigcap_{u \in N_T^+(z)} N^-(X_u)$; $L^o := L^o \cup \{z\}$.
    end while
    Return X_t.
end if

The algorithm **tree-Bipartition** is a subroutine used during the execution of **find-tree**. Let $T_1, \ldots, T_q$ be the subtrees of $T - v^*$, where v^* is a splitting vertex of the current call to **find-tree**. At the end of **tree-Bipartition**, we obtain a partition of the subtrees, or more precisely, a partition (WH, BL) of the indices $\{1, \ldots, q\}$ of the subtrees. The attained partition (WH, BL) is 'a greedily optimal bipartition' in certain sense while a nonnegative integer weight on an element of $\{1, \ldots, q\}$ is set to be $w(T_i)$ with some fine-tuning.

Lemma 2.5. *Consider the algorithm* **tree-Bipartition** *and let* (WH, BL) *be a bipartition of*

Algorithm 4 tree-Bipartition(T, t, v^*, L)

1: $T_1, \ldots, T_q$ are the subtrees of $T - v^*$. $Q := \{1, \ldots, q\}$. $w(T_i) := |V(T_i) \setminus L|$, $\forall i \in Q$.
2: **if** $v^* = t$ **then**
3: (A, B):=**Bipartition**$(Q, \{n_i := w(T_i) : i \in Q\})$
4: **if** $w(A) \leq w(B)$ **then** $WH := A$, $BL := B$. **else** $WH := B$, $BL := A$.
5: **else**
6: Let l be such that $t \in V(T_l)$
7: **if** $w(T_l) - w(v^*) \geq 0$ **then**
8: (A, B):=**Bipartition**$(Q, \{n_i := w(T_i) : i \in Q \setminus \{l\}\} \cup \{n_l := w(T_l) - w(v^*)\})$
9: **if** $l \in B$ **then** $WH := A$, $BL := B$. **else** $WH := B$, $BL := A$
10: **else** $\{w(T_l) - w(v^*) < 0\}$
11: (A, B):=**Bipartition**$((Q \setminus \{l\}) \cup \{v^*\}, \{n_i := w(T_i) : i \in Q \setminus \{l\}\} \cup \{n_{v^*} := w(v^*)\})$
12: **if** $v^* \in A$ **then** $WH := A - \{v^*\}$, $BL := B \cup \{l\}$. **else** $WH := B - \{v^*\}$, $BL := A \cup \{l\}$
13: **end if**
14: **end if**
15: Return (WH, BL).

$\{1, \ldots, q\}$ *obtained at the end of the algorithm. Then the partition* $U_w := \bigcup_{i \in WH} V(T_i) \cup \{v^*\}$ *and* $U_b := \bigcup_{i \in BL} V(T_i)$ *of* $V(T)$ *has the the following property.*
1) If $v^* = t$*, moving a component* T_i *from one partite set to the other does not decrease the difference* $d(w(U_w), w(U_b))$*.*
2) If $v^* \neq t$*, either exchanging* v^* *and the component* T_l *or moving a component* T_i*,* $i \neq v^*, l$ *from one partite set to the other does not decrease the difference* $d(w(U_w), w(U_b))$*.*

Proof. Let us consider the property 1). The bipartition (WH, BL) is determined in the first 'if' statement in line 2 of **tree-Bipartition**. Then by Lemma 2.3 the bipartition (WH, BL) is greedily optimal, which is equivalent to the statement of 1).

Let us consider the property 2). First suppose that the bipartition (WH, BL) is determined in the 'if' statement in line 7 of **tree-Bipartition**. The exchange of v^* and the component T_l amounts to moving the element l in the algorithm **Bipartition**. Since (WH, BL) is returned by **Bipartition** and thus is a greedily optimal bipartition of Q, any move of an element in one partite set would not decrease the difference $d(WH, BL)$ and the statement of 2) holds in this case.

Secondly suppose that the bipartition (WH, BL) is determined in the 'if' statement in line 10 of **tree-Bipartition**. In this case we have $w(T_l) = 0$ and thus exchanging T_l and v^* and amounts to moving the element v^* in the algorithm **Bipartition**. By the same argument as above, any move of an element in one partite set would not decrease the difference $d(WH, BL)$ and again the statement of 2) holds. □

Consider the following equation:

$$\alpha^2 - 3\alpha + 1 = 0 \tag{2}$$

Let $\alpha^* := (3 - \sqrt{5})/2$ be one of its roots. In line 10 of the algorithm **find-tree**, if $\alpha < \alpha^*$ we decide to pass the present splitting vertex v^* as a splitting vertex to the next recursive call which gets, as an argument, a subtree with greater weight among the two subtrees $T[U_w]$ and $T[U_b \cup \{v^*\}]$. Lemma 2.6 justifies this execution. It claims that if $\alpha < \alpha^*$, then in the next recursive call with a subtree of weight $(1-\alpha)w(T)$, we have a more balanced bipartition with v^* as a splitting vertex. Actually, the bipartition in the next step is good enough so as to compensate for the increase in the running time incurred by the biased ('$\alpha < \alpha^*$') bipartition in the present step. We will show this later.

Lemma 2.6. *Suppose that v^* has been chosen to split T for the present call to* **find-tree** *such that the weight of every subtree of $T - v^*$ is at most $w(T)/2$ and that $w(T) \geq 5$. Let α be defined as in line 8 and assume that $\alpha < \alpha^*$. Let $\{U_1, U_2\} = \{U_w, U_b\}$ such that $w(U_2) \geq w(U_1)$ and let $\{T_1, T_2\} = \{T[U_w], T[U_b \cup \{v^*\}]\}$ such that $U_1 \subseteq V(T_1)$ and $U_2 \subseteq V(T_2)$. Let α' play the role of α in the recursive call using the tree T_2. In this case the following holds: $\alpha' \geq (1-2\alpha)/(1-\alpha) > \alpha^*$.*

Proof. Let $T_1, T_2, U_1, U_2, \alpha, \alpha'$ be defined as in the statement. Note that $\alpha = w(U_1)/w(T)$. Let $d = w(U_2) - w(U_1)$ and note that $w(U_1) = (w(T) - d)/2$ and that the following holds

$$\frac{1-2\alpha}{1-\alpha} = \frac{w(T) - 2w(U_1)}{w(T) - w(U_1)} = \frac{2d}{w(T)+d}.$$

We now consider the following cases.

Case 1. $d = 0$: In this case $\alpha = 1/2 > \alpha^*$, a contradiction.

Case 2. $d = 1$: In this case $\alpha^* > \alpha = w(U_1)/(2w(U_1)+1)$, which implies that $w(U_1) \leq 1$. Therefore $w(U_2) \leq 2$ and $w(T) \leq 3$, a contradiction.

Case 3. $d \geq 2$: Let $C_1, C_2, \ldots, C_q$ denote the components in $T - v^*$ and without loss of generality assume that $V(C_1) \cup V(C_2) \cup \cdots \cup V(C_a) = U_2$ and $V(C_{a+1}) \cup V(C_{a+2}) \cup \cdots \cup V(C_q) = U_1$. Note that by Lemma 2.5 we must have $w(C_i) \geq d$ or $w(C_i) = 0$ for all $i = 1, 2, \ldots, q$ except possibly for one set C_l (containing t), which may have $w(C_l) = 1$ (if $w(v^*) = 1$).

Let C_r be chosen such that $w(C_r) \geq d$, $1 \leq r \leq a$ and $w(C_r)$ is minimum possible with these constraints. We first consider the case when $w(C_r) > w(U_2) - w(C_r)$. By the above (and the minimality of $V(C_r)$) we note that $w(U_2) \leq w(C_r)+1$ (as either C_j, which is defined above, or v^* may belong to $V(T_2)$, but not both). As $w(U_2) = (w(T)+d)/2 \geq w(T)/2+1$ we note that $w(C_r) \geq w(T)/2+d/2-1$. As $w(C_r) \leq w(T)/2$ (By the statement in our theorem) this implies that $d = 2$ and $w(C_r) = w(T)/2$ and $w(U_2) = w(C_r)+1$. If U_1 contains at least two distinct components with weight at least d then $w(U_1) > w(U_2)$, a contradiction. If U_1 contains no component of weight at least d then $w(U_1) \leq 1$ and $w(T) \leq 4$, a contradiction. So U_1 contains exactly one component of weight at least d. By the minimality of $w(C_r)$ we note that $w(U_1) \geq w(C_r) = w(U_2) - 1$, a contradiction to $d \geq 2$.

Therefore we can assume that $w(C_r) \leq w(U_2) - w(C_r)$, which implies the following (the last equality is proved above)

$$\alpha' \geq \frac{w(C_r)}{w(U_2)} \geq \frac{d}{(w(T)+d)/2} = \frac{1-2\alpha}{1-\alpha}.$$

As $\alpha < \alpha^*$, we note that $\alpha' \geq (1-2\alpha)/(1-\alpha) > (1-2\alpha^*)/(1-\alpha^*) = \alpha^*$. □

For the selection of the splitting vertex v^* we have two criteria in the algorithm **find-tree**: (i) *'found'* criterion: the vertex is found so that the weight of every subtree T' of $T - v^*$ is at most $w(T)/2$. (ii) *'taken-over'* criterion: the vertex is passed on to the present step as the argument v by the previous step of the algorithm. The following statement is an easy consequence of Lemma 2.6.

Corollary 2.7. *Suppose that $w(T) \geq 5$. If v^* is selected with 'taken-over' criterion, then $\alpha > \alpha^*$.*

Proof. For the initial call find-tree$(T, D, \emptyset, r, \emptyset, \emptyset)$ we have $v = \emptyset$ and thus, the splitting vertex v^* is selected with the 'found' criterion. We will prove the claim by induction. Consider the first vertex v^* selected with then 'taken-over' criterion during the performance of the algorithm. Then in the previous step, the splitting vertex was selected with 'found' criterion and thus in the present step we have $\alpha > \alpha^*$ by Lemma 2.6.

Now consider a vertex v^* selected with the 'taken-over' criterion. Then in the previous step, the splitting vertex was selected with the 'found' criterion since otherwise, by the induction hypothesis we have $\alpha > \alpha^*$ in the previous step, and $\emptyset$ has been passed on as the argument v for the present step. This is a contradiction. $\square$

Due to Corollary 2.7 the vertex v^* selected in line 3 of the algorithm **find-tree** functions properly as a splitting vertex. In other words, we have more than one subtree of $T - v^*$ in line 4 with positive weights.

Lemma 2.8. *If $w(T) \geq 2$, then for each of U_w and U_b found in line 5 of by* **find-tree** *we have $w(U_w) > 0$ and $w(U_b) > 0$.*

Proof. For the sake of contradiction suppose that one of $w(U_w)$ and $w(U_b)$ is zero. Let us assume $w(U_w) = 0$ and $w(U_b) = w(T)$. If v^* is selected with 'found' criteria, each component in $T[U_b]$ has a weight at most $w(T)/2$ and $T[U_b]$ contains at least two components of positive weights. Then we can move one component with a positive weight from U_b to U_w which will reduce the difference $d(U_w, U_b)$, a contradiction. The same argument applies when $w(U_w) = w(T)$ and $w(U_b) = 0$.

Consider the case when v^* is selected with "taken-over" criteria. There are three possibilities.

Case 1. $w(T) \geq 5$*:* In this case we obtain a contradiction with Corollary 2.7.

Case 2. $w(T) = 4$*:* In the previous step using T', where $T \subseteq T'$, the splitting vertex v^* was selected with "found" criteria. Then by the argument in the first paragraph, we have $w(T') \geq 5$. A contradiction follows from Lemma 2.6.

Case 3. $2 \leq w(T) \leq 3$*:* First suppose that $w(v^*) = 0$. Note that $T[U_w] - v^*$ or $T[U_b]$ contains a component of weight $w(T)$ since otherwise we can move a component with a positive weight from one partite set to the other and reduce $d(U_w, U_b)$. Considering the previous step using T', where $T \subseteq T'$, the out-tree T is the larger of T'_w and T'_b. We pass the splitting vertex v^* to the larger of the two only when $\alpha > \alpha^*$. So when $w(T) = 3$, we have $3 > (1 - \alpha^*)w(T')$ and thus $w(T') \leq 4$, and when $w(T) = 2$ we have $2 > (1 - \alpha^*)w(T')$ and thus $w(T') \leq 3$. In either case, however, $T' - v^*$ contains a component with a weight greater than $w(T')/2$, contradicting to the choice of v^* in the previous step (Recall that v^* is selected with 'found' criteria in the previous step using T').

Secondly suppose that that $w(v^*) = 1$. Then $w(U_w) = w(T)$ and $w(U_b) = 0$. We can reduce the difference $d(U_w, U_b)$ by moving the component with a positive weight from U_w to U_b, a contradiction.

Therefore for each of U_w and U_b found in line 5 of by **find-tree** we have $w(U_w) > 0$ and $w(U_b) > 0$. $\square$

Lemma 2.9. *Given a digraph D, an out-tree T and a specified vertex $t \in V(T)$, consider the set X_t (in line 22) returned by the algorithm* **find-tree**$(T, D, v, t, L, \{X_u : u \in L\})$*. We assume that the sets X_u, $u \in L$ are pairwise disjoint. If $w \in X_t$ then D contains a (t, w)-tree that meets the restrictions on L. Conversely, if D contains a (t, w)-tree for a vertex $w \in V(D)$ that meets the restrictions on L, then X_t contains w with probability larger than $1 - 1/e > 0.6321$.*

Proof. Lemma 2.8 guarantees that the splitting vertex v^* selected at any recursive call of **find-tree** really 'splits' the input out-tree T into two nontrivial parts, unless $w(T) \leq 1$.

First we show that if $w \in X_t$ then D contains a (t, w)-tree for a vertex $w \in V(D)$ that meets the restrictions on L. When $|V(T) \setminus L| \leq 1$, using Lemma 2.4 it is straightforward to check from the algorithm that the claim holds. Assume that the claim is true for all subsequent calls to **find-tree**. Since $w \in S'$ for some S' returned by a call in line 18, the subgraph $D[V_b \cup X_{v^*}]$ contains a $T[U_b \cup \{v^*\}]$-isomorphic (t, w)-tree T_D^b meeting the restrictions on $(L \cap U_b) \cup \{v^*\}$ by induction hypothesis. Moreover, $X_{v^*} \neq \emptyset$ when $S' \ni w$ is returned and this implies that there is a vertex $u \in X_{v^*}$ such that T_D^b is a (v^*, u)-tree. Since $u \in X_{v^*}$, induction hypothesis implies that the subgraph $D[V_w]$ contains a $T[U_w]$-isomorphic (v^*, u)-tree, say T_D^w.

Consider the subgraph $T_D := T_D^w \cup T_D^b$. To show that T_D is a T-isomorphic (t, w)-tree in D, it suffices to show that $V(T_D^w) \cap V(T_D^b) = \{u\}$. Indeed, $V(T_D^w) \subseteq V_w$, $V(T_D^b) \subseteq V_b \cup X_{v^*}$ and $V_w \cap V_b = \emptyset$. Thus if two trees T_D^w and T_D^b share vertices other than u, these common vertices should belong to X_{v^*}. Since T_D^b meets the restrictions on $(L \cap U_b) \cup \{v^*\}$, we have $X_{v^*} \cap V(T_D^b) = \{u\}$. Hence u is the only vertex that two trees T_D^w and T_D^b have in common. We know that u plays the role of v^* in both trees. Therefore we conclude that T_D is T-isomorphic, and since w plays the role of t, it is a (t, w)-tree. Obviously T_D meets the restrictions on L.

Secondly, we shall show that if D contains a (t, w)-tree for a vertex $w \in V(D)$ that meets the restrictions on L, then X_t contains w with probability larger than $1 - 1/e > 0.6321$. When $|V(T) \setminus L| \leq 1$, the algorithm **find-tree** is deterministic and returns X_t which is exactly the set of all vertices w for which there exists a (t, w)-tree meeting the restrictions on L. Hence the claim holds for the base case, and we may assume that the claim is true for all subsequent calls to **find-tree**.

Suppose that there is a (t, w)-tree T_D meeting the restrictions on L and that this is a (v^*, w')-tree, that is, the vertex w' plays the role of v^*. Then the vertices of T_D corresponding to U_w, say T_D^w, are colored white and those of T_D corresponding to U_b, say T_D^b, are colored black as intended with probability $\geq (\alpha^\alpha(1-\alpha)^{1-\alpha})^k$. When we hit the right coloring for T, the digraph $D[V_w]$ contains the subtree T_D^w of T_D which is $T[U_w]$-isomorphic and which is a (v^*, w')-tree. By induction hypothesis, the set S obtained in line 15 contains w' with probability larger than $1 - 1/e$. Note that T_D^w meets the restrictions on $L \cap U_w$.

If $w' \in S$, the restrictions delivered onto the subsequent call for **find-tree** in line 17 contains w'. Since T_D meets the restrictions on L confined to $U_b - v^*$ and it is a (v^*, w')-tree with $w' \in S = X_{v^*}$, the subtree T_D^b of T_D which is $T[U_b \cup \{v^*\}]$-isomorphic meets all the restrictions on L. Hence by induction hypothesis, the set S' returned in line 18 contains w with probability larger than $1 - 1/e$.

The probability ρ that S', returned by **find-tree** in line 18 at an iteration of the loop, contains w is, thus,

$$\rho > (\alpha^\alpha(1-\alpha)^{1-\alpha})^k \times (1 - 1/e)^2 > 0.3995(\alpha^\alpha(1-\alpha)^{1-\alpha})^k.$$

After looping $\lceil (0.3995(\alpha^\alpha(1-\alpha)^{1-\alpha})^k)^{-1} \rceil$ times in line 12, the probability that X_t contains w is at least

$$1-(1-\rho)^{1/(0.3995(\alpha^\alpha(1-\alpha)^{1-\alpha})^k)} > 1-(1-0.3995(\alpha^\alpha(1-\alpha)^{1-\alpha})^k)^{1/(0.3995(\alpha^\alpha(1-\alpha)^{1-\alpha})^k)} > 1-\frac{1}{e}.$$

Observe that the probability ρ does not depend on α and the probability of coloring a vertex white/black. $\square$

The complexity of Algorithm **find-tree** is analyzed in the following theorem.

Theorem 2.10. *Algorithm* **find-tree** *has running time* $O(n^2k^\rho C^k)$*, where* $w(T) = k$ *and* $|V(D)| = n$*, and* C *and* ρ *are defined and bounded as follows:*

$$C = \left(\frac{1}{\alpha^{*\alpha^*}(1-\alpha^*)^{1-\alpha^*}}\right)^{1/\alpha^*}, \ \rho = \frac{\ln(1/6)}{\ln(1-\alpha^*)}, \ \rho \leq 3.724, \ \textit{and } C \leq 5.7039.$$

Proof. Let $L(T,D)$ denote the number of times the 'if'-statement in line 1 of Algorithm **find-tree** is false (in all recursive calls to **find-tree**). We will prove that $L(T,D) \leq R(k) = Bk^\rho C^k + 1$, $B \geq 1$ is a constant whose value will determined later in the proof. This would imply that the number of calls to **find-tree** where the 'if'-statement in line 1 is true is also bounded by $R(k)$ as if line 1 is true then we will have at least two calls to **find-tree** (in fact it will have at least three as $\left\lceil \frac{2.51}{\alpha^{\alpha k}(1-\alpha)^{(1-\alpha)k}} \right\rceil \geq 3$ and we always have a call in line 15). We can therefore think of the search tree of Algorithm 3 as an out-tree where all internal nodes have out-degree at least two and therefore the number of leaves is greater than the number of internal nodes.

Observe that each iteration of the for-loop in line 12 of Algorithm **find-tree** makes at most two recursive calls to **find-tree** and the time spent in each iteration of the for-loop is at most $O(n^2)$. As the time spent in each call of **find-tree** outside the for-loop is also bounded by $O(n^2)$ we obtain the desired complexity bound $O(n^2k^\rho C^k)$.

Thus, it remains to show that $L(T,D) \leq R(k) = Bk^\rho C^k + 1$. First note that if $k = 0$ or $k = 1$ then line 1 is false exactly once (as there are no recursive calls) and $\min\{R(1), R(0)\} \geq 1 = L(T,D)$. If $k \in \{3,4\}$, then line 1 is false a constant number of times by Lemma 2.8 and let B be the minimal integer such that $L(T,D) \leq R(k) = Bk^\rho C^k + 1$ for both $k = 3$ and 4. Thus, we may now assume that $k \geq 5$ and proceed by induction on k.

Let $R'(\alpha,k) = (6((1-\alpha)k)^\rho C^{(1-\alpha)k})/(\alpha^{\alpha k}(1-\alpha)^{(1-\alpha)k})$. Let α be defined as in line 8 of Algorithm **find-tree**. We will consider the following two cases separately.

Case 1, $\alpha \geq \alpha^*$*:* In this case we note that the following holds as $k \geq 2$ and $(1-\alpha) \geq \alpha$.

$$\begin{aligned} L(T,D) &\leq \left\lceil 2.51/(\alpha^{\alpha k}(1-\alpha)^{(1-\alpha)k}) \right\rceil \times (R(\alpha k) + R((1-\alpha)k)) \\ &\leq 3/(\alpha^{\alpha k}(1-\alpha)^{(1-\alpha)k}) \times (2 \cdot R((1-\alpha)k)) \\ &= R'(\alpha,k). \end{aligned}$$

By the definition of ρ we observe that $(1-\alpha^*)^\rho = 1/6$, which implies that the following holds by the definition of C:

$$R'(\alpha^*,k) = 6((1-\alpha^*)k)^\rho C^{(1-\alpha^*)k} \times C^{\alpha^* k} = k^\rho C^k = R(k).$$

Observe that

$$\ln(R'(\alpha,k)) = \ln(6) + \rho\left[\ln(k) + \ln(1-\alpha)\right] + k\left[(1-\alpha)\ln(C) - \alpha\ln(\alpha) - (1-\alpha)\ln(1-\alpha)\right]$$

We now differentiate $\ln(R'(\alpha,k))$ which gives us the following:

$$\begin{aligned} \frac{\partial(\ln(R'(\alpha,k)))}{\partial(\alpha)} &= \rho\frac{-1}{1-\alpha} + k\left(-\ln(C) - (1+\ln(\alpha)) + (1+\ln(1-\alpha))\right) \\ &= \frac{-\rho}{1-\alpha} + k\left(\ln\left(\frac{1-\alpha}{\alpha C}\right)\right). \end{aligned}$$

Since $k \geq 0$ we note that the above equality implies that $R'(\alpha, k)$ is a decreasing function in α in the interval $\alpha^* \leq \alpha \leq 1/2$. Therefore $L(T, D) \leq R'(\alpha, k) \leq R'(\alpha^*, k) = R(k)$, which proves Case 1.

Case 2, $\alpha < \alpha^*$: In this case we will specify the splitting vertex when we make recursive calls using the larger of U_w and U_b (defined in line 5 of Algorithm **find-tree**). Let α' denote the α-value in such a recursive call. By Lemma 2.6 we note that the following holds:

$$\frac{1}{2} \geq \alpha' \geq \frac{1-2\alpha}{1-\alpha} > \alpha^*.$$

Analogously to Case 1 (as $R'(\alpha', (1-\alpha)k)$ is a decreasing function in α' when $1/2 \geq \alpha' \geq \alpha^*$) we note that the L-values for these recursive calls are bounded by the following, where $\beta = \frac{1-2\alpha}{1-\alpha}$ (which implies that $(1-\alpha)(1-\beta) = \alpha$):

$$\begin{aligned}
R'(\alpha', (1-\alpha)k) &\leq R'\left(\beta, (1-\alpha)k\right) \\
&= 3/\left(\left(\beta^{\beta}(1-\beta)^{(1-\beta)}\right)^{(1-\alpha)k}\right) \times 2 \times R((1-\beta)(1-\alpha)k) \\
&= 6R(\alpha k)/\left(\left(\beta^{\beta}(1-\beta)^{(1-\beta)}\right)^{(1-\alpha)k}\right).
\end{aligned}$$

Thus, in the worst case we may assume that $\alpha' = \beta = (1-2\alpha)/(1-\alpha)$ in all the recursive calls using the larger of U_w and U_b. The following now holds (as $k \geq 2$).

$$\begin{aligned}
L(T, D) &\leq \left\lceil 2.51/(\alpha^{\alpha k}(1-\alpha)^{(1-\alpha)k}) \right\rceil \times (R(\alpha k) + R'(\alpha', (1-\alpha)k)) \\
&\leq 3/(\alpha^{\alpha k}(1-\alpha)^{(1-\alpha)k}) \times R(\alpha k) \times \left(1 + 6/\left(\left(\beta^{\beta}(1-\beta)^{(1-\beta)}\right)^{(1-\alpha)k}\right)\right) \\
&\leq 3R(\alpha k)/(\alpha^{\alpha k}(1-\alpha)^{(1-\alpha)k}) \times 7/\left(\left(\beta^{\beta}(1-\beta)^{(1-\beta)}\right)^{(1-\alpha)k}\right)
\end{aligned}$$

Let $R^*(\alpha, k)$ denote the bottom right-hand side of the above equality (for any value of α). By the definition of ρ we note that $\rho = \frac{2\ln(1/6)}{2\ln(1-\alpha^*)} = \frac{\ln(1/36)}{\ln(\alpha^*)}$, which implies that $(\alpha^*)^{\rho} = 1/36$. By the definition of C and the fact that if $\alpha = \alpha^*$ then $\beta = (1-2\alpha^*)/(1-\alpha^*) = \alpha^*$, we obtain the following:

$$\begin{aligned}
R^*(\alpha^*, k) &= 3R(\alpha^* k)/(\alpha^{*\alpha^* k}(1-\alpha^*)^{(1-\alpha^*)k}) \times 7/\left(\left(\alpha^{*\alpha^*}(1-\alpha^*)^{(1-\alpha^*)}\right)^{(1-\alpha^*)k}\right) \\
&= 21 \cdot R(\alpha^* k) \cdot C^{\alpha^* k} \cdot C^{\alpha^*(1-\alpha^*)k} \\
&= 21\alpha^{*\rho} k^{\rho} C^{\alpha^* k} \times C^{(2\alpha^* - \alpha^{*2})k} \\
&= 21\alpha^{*\rho} R(k) \\
&< R(k).
\end{aligned}$$

We will now simplify $R^*(\alpha, k)$ further, before we differentiate $\ln(R^*(\alpha, k))$. Note that $\beta = \frac{1-2\alpha}{1-\alpha}$ implies that $(1-\alpha)(1-\beta) = \alpha$ and $\beta(1-\alpha) = 1-2\alpha$.

$$\begin{aligned}
R^*(\alpha, k) &= 21R(\alpha k)/(\alpha^{\alpha k}(1-\alpha)^{(1-\alpha)k}) \times 1/\left(\left(\beta^{\beta}(1-\beta)^{(1-\beta)}\right)^{(1-\alpha)k}\right) \\
&= 21(\alpha k)^{\rho} C^{\alpha k}/(\alpha^{\alpha k}(1-\alpha)^{(1-\alpha)k}) \times 1/\left(\left(\tfrac{1-2\alpha}{1-\alpha}\right)^{(1-2\alpha)k}\left(\tfrac{\alpha}{1-\alpha}\right)^{\alpha k}\right) \\
&= 21(\alpha k)^{\rho}\left(C^{\alpha}/(\alpha^{2\alpha}(1-2\alpha)^{(1-2\alpha)})\right)^{k}.
\end{aligned}$$

Thus, we have the following:

$$\ln(R^*(\alpha,k)) = \ln(21) + \rho\left(\ln(k) + \ln(\alpha)\right) + k\left(\alpha\ln(C) - 2\alpha\ln(\alpha) - (1-2\alpha)\ln(1-2\alpha)\right).$$

We now differentiate $\ln(R^*(\alpha,k))$ which gives us the following:

$$\begin{aligned}\frac{\partial(\ln(R^*(\alpha,k)))}{\partial(\alpha)} &= \frac{\rho}{\alpha} + k\left(\ln(C) - 2(1+\ln(\alpha)) + 2(1+\ln(1-2\alpha))\right)\\ &= \frac{\rho}{\alpha} + k\left(\ln\left(\frac{C(1-2\alpha)^2}{\alpha^2}\right)\right)\end{aligned}$$

Since $k \geq 0$ we note that the above equality implies that $R^*(\alpha,k)$ is an increasing function in α in the interval $1/3 \leq \alpha \leq \alpha^*$. Therefore $L(T,D) \leq R^*(\alpha,k) \leq R^*(\alpha^*,k) < R(k)$, which proves Case 2. □

Theorem 2.11. *There is an $O(n^2 5.704^k)$ time randomized algorithm that solves the k-*Out-Tree *problem.*

2.3 Derandomization of Our Randomized Algorithm for k-Out-Tree

In this subsection we discuss the derandomization of the algorithm **find-tree** using the general method presented by Chen et al. [4] and based on the construction of (n,k)-universal sets studied in [11].

Definition 2.12. *An (n,k)-universal set $\mathcal{F}$ is a set of functions from $[n]$ to $\{0,1\}$, such that for every subset $S \subseteq [n], |S| = k$ the set $\mathcal{F}|_S = \{f|_S : f \in \mathcal{F}\}$ is equal to the set 2^S of all the functions from S to $\{0,1\}$.*

Such an universal set can play in **find-tree** the role of the random colorings. In the same article [4], Chen et al. also give an algorithm to generate one :

Proposition 2.13. *([4]) There is an $O(n2^{k+12\log^2 k})$ time deterministic algorithm that constructs an (n,k)-universal set of size bounded by $n2^{k+12\log^2 k+2}$.*

Using this universal set alone, however, would not enable us to obtain a deterministic fixed-parameter algorithm for **find-tree**, as the size of the family (and, thus, the number of iterations in the main loop of the algorithm) would now also depend on n, besides k. Hence, Chen et al. make use (see [4]) of a family of pre-coloring functions $(g_{n,k,z})_{z\leq 2n}$ to obtain a fixed-parameter algorithm. To explain it, let us first give a result from Fredman et al. [8].

Proposition 2.14. *Let n and k be integers, $n \geq k$, and let q_0 be the smallest prime number such that $n \leq q_0 < 2n$. For any k-subset S in $Z_n = 0,\ldots,n-1$, there is an integer z, $0 \leq z \leq q_0$, such that the function $g_{n,k,z}$ over Z_n, defined as $g_{n,k,z}(a) = (az \mod q_0) \mod k^2$, is injective from S.*

By the above proposition, computing a (k^2,k)-universal set $\mathcal{F}_{k^2,k}$ instead of a (n,k)-universal set is enough for our purposes. Indeed, if we are looking for a k-subgraph S in our graph, there exists $1 \leq z \leq 2n$ such that $g_{n,k,z}$ is injective on S, thus ensuring that the family $\mathcal{F}'_{k,n,z} = \mathcal{F}_{k^2,k} \circ g_{n,k,z} = \{f \circ g_{n,k,z}, f \in \mathcal{F}_{k^2,k}\}$ is such that $\mathcal{F}'_{k,n,z}|_S$ is equal to the set 2^S.

This way, derandomizing **find-tree** amounts to running it at most $2n$ times (once for each possible value of z), each time using as a set of coloring functions the family $\mathcal{F}'_{k,n,z}$. Two lines of the algorithm will then need to be modified :

13 **for** each function $f \in F'_{k,n,z}$ **do**

14 $\forall i$ such that $x_i \in V(D) - \bigcup_{u \in L} X_u$, let v_i be colored in white if $f(i) = 0$ and in black if $f(i) = 1$

Besides, we also need to pre-compute a (k^2, k')-universal set for any $k' \leq k$, as this will be needed in the recursions steps of the algorithm. By Proposition 2.13, this can be done in time $O(k^3 2^{k+12\log^2 k})$. Note that these modifications make the algorithm **find-tree** deterministic.

Then, from Proposition 2.14 we deduce that if a digraph D contains an out-tree T meeting the requirements, then there exists a z such that $g_{n,k,z}$ is injective on $V(T)$. During the iteration of the algorithm corresponding to z there will be an $f \in \mathcal{F}'_{k,n,z}$ such that the vertices corresponding to U_w in D with be colored in white while the vertices corresponding to U_b will be colored in black. Using induction on k, we can prove that this deterministic algorithm correctly returns the required out-tree provided that such an out-tree exists in the digraph.

Let us briefly sketch how the running time is derived. We consider the following type of recurrence relations:

$$T(k,n) \leq X_0 2^k \times (T((1-\alpha)k, n) + T(\alpha k, n))$$

Here X_0 is a constant determined by the size of the initial out-tree we are considering, and it adds to the exponent of $T(k,n)$ with $o(k)$ factor. On the other hand, the value of α asymtotically evolves around α^* as we see in the randomized version of algorithm. As a result, $T(k,n)$ is a function of the form $(2^{1/\alpha^*})^{k+o(k)}$. Overall the computation is similar to that described in the proof of Theorem 2.10. Thus, we obtain the following:

Theorem 2.15. *There is an $O(n^2 6.139^{k+o(k)}) = O(n^2 6.14^k)$ time deterministic algorithm that solves the k-Out-Tree problem.*

3 Algorithm for k-Int-Out-Branching

A *k-internal out-tree* is an out-tree with at least k internal vertices. We call a k-internal out-tree *minimal* if none of its proper subtrees is a k-internal out-tree, or *minimal k-tree* in short. The Rooted Minimal k-Tree problem is as follows: given a digraph D, a vertex u of D and a minimal k-tree T, where k is a parameter, decide whether D contains an out-tree rooted at u and isomorphic to T. Recall that k-Int-Out-Branching is the following problem: given a digraph D and a parameter k, decide whether D contains an out-branching with at least k internal vertices. Finally, the k-Int-Out-Tree problem is stated as follows: given a digraph D and a parameter k, decide whether D contains an out-tree with at least k internal vertices.

Lemma 3.1. *Let T be a k-internal out-tree. Then T is minimal if and only if $|Int(T)| = k$ and every leaf $u \in Leaf(T)$ is the only child of its parent $N^-(u)$.*

Proof. Assume that T is minimal. It cannot have more than k internal vertices, because otherwise by removing any of its leaves, we obtain a subtree of T with at least k internal vertices. Thus $|\mathrm{Int}(T)| = k$. If there are sibling leaves u and w, then removing one of them provides a subtree of T with $|\mathrm{Int}(T)|$ internal vertices.

Now, assume that $|\mathrm{Int}(T)| = k$ and every leaf $u \in \mathrm{Leaf}(T)$ is the only child of its parent $N^-(u)$. Observe that every subtree of T can be obtained from T by deleting a leaf of T, a leaf in the resulting out-tree, etc. However, removing any leaf v from T decreases the number of internal vertices, and thus creates subtrees with at most $k-1$ internal vertices. Thus, T is minimal.

$\square$

In fact, Lemma 3.1 can be used to generate all non-isomorphic minimal k-trees. First, build an (arbitrary) out-tree T^0 with k vertices. Then extend T^0 by adding a vertex x' for each leaf $x \in \mathrm{Leaf}(T^0)$ with an arc (x, x'). The resulting out-tree T' satisfies the properties of Lemma 3.1. Conversely, by Lemma 3.1, any minimal k-tree can be constructed in this way.

Generating Minimal k-Tree (GMT) Procedure
a. Generate a k-vertex out-tree T^0 and a set $T' := T^0$.
b. For each leaf $x \in \mathrm{Leaf}(T')$, add a new vertex x' and an arc (x, x') to T'.

Due to the following simple observation, to solve k-Int-Out-Tree for a digraph D it suffices to solve Rooted Minimal k-Tree for each vertex $u \in V(D)$ and each minimal k-tree T rooted at u.

Lemma 3.2. *Any k-internal out-tree rooted at r contains a minimal k-tree rooted at r as a subdigraph.*

Similarly, the next two lemmas show that to solve k-Out-Branching for a digraph D it suffices to solve Rooted Minimal k-Tree for each vertex $u \in S$ and each minimal k-tree T rooted at u, where S is the unique strong connectivity component of D without incoming arcs.

Lemma 3.3. *[2] A digraph D has an out-branching rooted at vertex $r \in V(D)$ if and only if D has a unique strong connectivity component S of D without incoming arcs and $r \in S$. One can check whether D has a unique strong connectivity component and find one, if it exists, in time $O(m+n)$, where n and m are the number of vertices and arcs in D, respectively.*

The next lemma is a folklore.

Lemma 3.4. *Suppose a given digraph D with n vertices and m arcs has an out-branching rooted at vertex r. Then any minimal k-tree rooted at r can be extended to a k-internal out-branching rooted at r in time $O(m+n)$.*

Since k-Int-Out-Tree and k-Int-Out-Branching can be solved similarly, we will only deal with the k-Int-Out-Branching problem. We will assume that our input digraph contains a unique strong connectivity component S. Our algorithm called *IOBA* for solving k-Int-Out-Branching for a digraph D runs in two stages. In the first stage, we generate *all* minimal k-trees. We use the GMT procedure described above to achieve this. At the second stage, for each $u \in S$ and each minimal k-tree T, we check whether D contains an out-tree rooted at u and isomorphic to T using our algorithm from the previous section. We return TRUE if and only if we succeed in finding an out-tree H of D rooted at $u \in S$ which is isomorphic to a minimal k-tree.

In the literature, mainly rooted (undirected) trees and not out-trees are studied. However, every rooted tree can be made an out-tree by orienting every edge away from the root and every out-tree can be made a rooted tree by disregarding all orientations. Thus, rooted trees and out-trees are equivalent and we can use results obtained for rooted trees for out-trees.

Otter [13] showed that the number of non-isomorphic out-trees on k vertices is $t_k = O^*(2.95^k)$. We can generate all non-isomorphic rooted trees on k vertices using the algorithm of Beyer and Hedetniemi [3] of runtime $O(t_k)$. Using the GMT procedure we generate all minimal k-trees. We see that the first stage of IOBA can be completed in time $O^*(2.95^k)$.

In the second stage of IOBA, we try to find a copy of a minimal k-tree T in D using our algorithm from the previous section. The running time of our algorithm is $O^*(6.14^k)$. Since the number of vertices of T is bounded from above by $2k-1$, the overall running time for the second stage of the algorithm is $O^*(2.95^k \cdot 6.14^{2k-1})$. Thus, the overall time complexity of the algorithm is $O^*(2.95^k \cdot 6.14^{2k-1}) = O^*(112^k)$.

We can reduce the complexity with a more refined analysis of the algorithm. The major contribution to the large constant 112 in the above simple analysis comes from the running time of our algorithm from the previous section. There we use the upper bound on the number of vertices in a minimal k-tree. Most of the minimal k-trees have less than $k-1$ leaves, which implies that the upper bound $2k-1$ on the order of a minimal k-tree is too big for the majority of the minimal k-trees. Let $T(k)$ be the running time of IOBA. Then we have

$$T(k) = O^*\left(\sum_{k+1 \le k' \le 2k-1} (\#\text{ of minimal } k\text{-trees on } k' \text{ vertices}) \times (6.14^{k'})\right) \tag{3}$$

A minimal k-tree T' on k' vertices has $k'-k$ leaves, and thus the out-tree T^0 from which T' is constructed has k vertices of which $k'-k$ are leaves. Hence the number of minimal k-trees on k' vertices is the same as the number of non-isomorphic out-trees on k vertices with $k'-k$ leaves. Here an interesting counting problem arises. Let $g(k,l)$ be the number of non-isomorphic out-trees on k vertices with l leaves. Enumerate $g(k,l)$. To our knowledge, such a function has not been studied yet. Leaving it as a challenging open question, here we give an upper bound on $g(k,l)$ and use it for a better analysis of $T(k)$. In particular we are interested in the case when $l \ge k/2$.

Consider an out-tree T^0 on $k \ge 3$ vertices which has αk internal vertices and $(1-\alpha)k$ leaves. We want to obtain an upper bound on the number of such non-isomorphic out-trees T^0. Let T^c be the subtree of T^0 obtained after deleting all its leaves and suppose that T^c has βk leaves. Assume that $\alpha \le 1/2$ and notice that αk and βk are integers. Clearly $\beta < \alpha$.

Each out-tree T^0 with $(1-\alpha)k$ leaves can be obtained by appending $(1-\alpha)k$ leaves to T^c so that each of the vertices in Leaf(T^c) has at least one leaf appended to it. Imagine that we have $\beta k = |\text{Leaf}(T^c)|$ and $\alpha k - \beta k = |\text{Int}(T^c)|$ distinct boxes. Then what we are looking for is the number of ways to put $(1-\alpha)k$ balls into the boxes so that each of the first βk boxes is nonempty. Again this is equivalent to putting $(1-\alpha-\beta)k$ balls into αk distinct boxes. It is an easy exercise to see that this number equals $\binom{k-\beta k-1}{\alpha k-1}$.

Note that the above number does not give the exact value for the non-isomorphic out-trees on k vertices with $(1-\alpha)k$ leaves. This is because we treat an out-tree T^c as a labeled one, which may lead to us to distinguishing two assignments of balls even though the two corresponding out-trees T^0's are isomorphic to each other.

A minimal k-tree obtained from T^0 has $(1-\alpha)k$ leaves and thus $(2-\alpha)k$ vertices. With the upper bound $O^*(2.95^{\alpha k})$ on the number of T^c's by [13], by (3) we have the following:

$$\begin{aligned}
T(k) &= O^*\left(\sum_{\alpha \le 1/2}\sum_{\beta<\alpha} 2.95^{\alpha k}\binom{k-\beta k-1}{\alpha k-1}(6.14)^{(2-\alpha)k}\right) + O^*\left(\sum_{\alpha>1/2} 2.95^{\alpha k}(6.14)^{(2-\alpha)k}\right)\\
&= O^*\left(\sum_{\alpha \le 1/2}\sum_{\beta<\alpha} 2.95^{\alpha k}\binom{k}{\alpha k}(6.14)^{(2-\alpha)k}\right) + O^*\left(2.95^{k}(6.14)^{3k/2}\right)\\
&= O^*\left(\sum_{\alpha \le 1/2}\left(2.95^{\alpha}\frac{1}{\alpha^{\alpha}(1-\alpha)^{1-\alpha}}(6.14)^{(2-\alpha)}\right)^k\right) + O^*(44.9^k)
\end{aligned}$$

The term in the sum over $\alpha \le 1/2$ above is maximized when $\alpha = \frac{2.95}{2.95+6.14}$, which yields $T(k) = O^*(55.8^k)$. Thus, we conclude with the following theorem.

Theorem 3.5. k-Int-Out-Branching *is solvable in time* $O^*(55.8^k)$.

4 Conclusion

In this paper we refine the approach of Chen et al. [4] based on randomized Divide-and-Conquer technique. Our technique is based on a more complicated coloring and within this technique we refined the result of Alon et al. [1] for the k-Out-Tree problem. It is interesting to see if this technique can be used to obtain faster algorithms for other parameterized problems.

As a byproduct of our work, we obtained the first $O^*(2^{O(k)})$-time algorithm for k-Int-Out-Branching. We used the classical result of Otter [13] that the number of non-isomorphic trees on k vertices is $O^*(2.95^k)$. An interesting combinatorial problem is to refine this bound for trees having $\lfloor \alpha k \rfloor$ leaves for some $\alpha < 1$.

References

[1] N. Alon, R. Yuster and U. Zwick, Color-coding. *Journal of the ACM* 42 (1995), 844–856.

[2] J. Bang-Jensen and G. Gutin, Digraphs: Theory, Algorithms and Applications, 2nd Ed., Springer-Verlag, London, 2008.

[3] T. Beyer and S.M. Hedetniemi, Constant time generation of rooted trees. *SIAM J. Computing* 9 (1980), 706–712.

[4] J. Chen, J. Kneis, S. Lu, D. Mölle, S. Richter, P. Rossmanith, S.-H. Sze, and F. Zhang, Randomized divide-and-conquer: Improved path, matching, and packing algorithms. *SIAM J. Computing* 38 (2009), 2526–2547.

[5] F. R. K. Chung, Separator theorems and their applications, In Paths, Flows, and VLSI-Layout (Bonn, 1988), Springer-Verlag, Berlin, pp. 17–34, 1990.

[6] N. Cohen, F. Fomin, G. Gutin, E.J. Kim, S. Saurabh and A. Yeo, Algorithm for Finding k-Vertex Out-trees and its Application to k-Internal Out-branching Problem. *Proc. COCOON'09*, Lect. Notes Comput. Sci. 5609 (2009), 37–46.

[7] A. Demers and A. Downing, Minimum leaf spanning tree. US Patent no. 6,105,018, August 2000.

[8] M.L. Fredman, J. Komlos and E. Szemeredi, Storing a sparse table with $O(1)$ worst case access time. *Journal of the ACM* 31 (1984), 538–544.

[9] G. Gutin, I. Razgon and E.J. Kim, Minimum Leaf Out-Branching Problems. *Proc. AAIM'08*, Lecture Notes Comput. Sci. 5034 (2008), 235–246.

[10] I. Koutis, Faster algebraic algorithms for path and packing problems. *Proc. ICALP'08*, Lecture Notes Comput. Sci. 5125 (2008), 575–586.

[11] M. Naor, L. J. Schulman and A. Srinivasan, Splitters and Near-Optimal Derandomization, Proc. 17th Ann. Symp. Found. Comput. Sci. (1995), 182–193

[12] A. Nilli, Perfect hashing and probability, *Combinatorics Prob. Comput.* 3 (1994), 407–409.

[13] R. Otter, The Number of Trees. Ann. Math. 49 (1948), 583–599.

[14] E. Prieto and C. Sloper, Either/Or: Using Vertex Cover Structure in desigining FPT-algorithms - The Case of k-Internal Spanning Tree, *Proc. WADS'2003*, Lect. Notes Comput. Sci. 2748 (2003), 465–483.

[15] E. Prieto and C. Sloper, Reducing To Independent Set Structure - The Case of k-Internal Spanning Tree, *Nordic Journal of Computing* 15 (2005), 308–318.

[16] R. Williams, Finding a path of length k in $O^*(2^k)$ time. *Inform. Proc. Letters* 109 (2009), 315–318.

Tradeoffs in process strategy games with application in the WDM reconfiguration problem

Nathann Cohen[a], David Coudert[a], Dorian Mazauric[a], Napoleão Nepomuceno[a,b], Nicolas Nisse[a]

[a]*MASCOTTE, INRIA, I3S, CNRS, Univ. Nice Sophia*
2004 route des Lucioles – B.P. 93 – 06902 Sophia Antipolis Cedex – France
[b]*Universidade Federal do Ceará, Fortaleza-CE, Brazil*

Abstract

We consider a variant of the graph searching games that models the routing reconfiguration problem in WDM networks. In the digraph processing game, a team of agents aims at *processing*, or clearing, the vertices of a digraph D. We are interested in two different measures: 1) the total number of agents used, and 2) the total number of vertices occupied by an agent during the processing of D. These measures respectively correspond to the maximum number of simultaneous connections interrupted and to the total number of interruptions during a reconfigurationrouting in a WDM network.

Previous works have studied the problem of independently minimizing each of these parameters. In particular, the corresponding minimization problems are APX-hard, and the first one is known not to be in APX. In this paper, we give several complexity results and study tradeoffs between these conflicting objectives. In particular, we show that minimizing one of these parameters while the other is constrained is NP-complete. Then, we prove that there exist some digraphs for which minimizing one of these objectives arbitrarily impairs the quality of the solution for the other one. We show that such bad tradeoffs may happen even for a basic class of digraphs. On the other hand, we exhibit classes of graphs for which good tradeoffs can be achieved. We finally detail the relationship between this game and the routing reconfiguration problem. In particular, we prove that any instance of the processing game, i.e. any digraph, corresponds to an instance of the routing reconfiguration problem.

Email addresses: `Nathann.Cohen@sophia.inria.fr` (Nathann Cohen), `David.Coudert@sophia.inria.fr` (David Coudert), `Dorian.Mazauric@sophia.inria.fr` (Dorian Mazauric), `Napoleao.Nepomuceno@sophia.inria.fr` (Napoleão Nepomuceno), `Nicolas.Nisse@sophia.inria.fr` (Nicolas Nisse)

Keywords: graph searching games ; process number ; routing reconfiguration problem

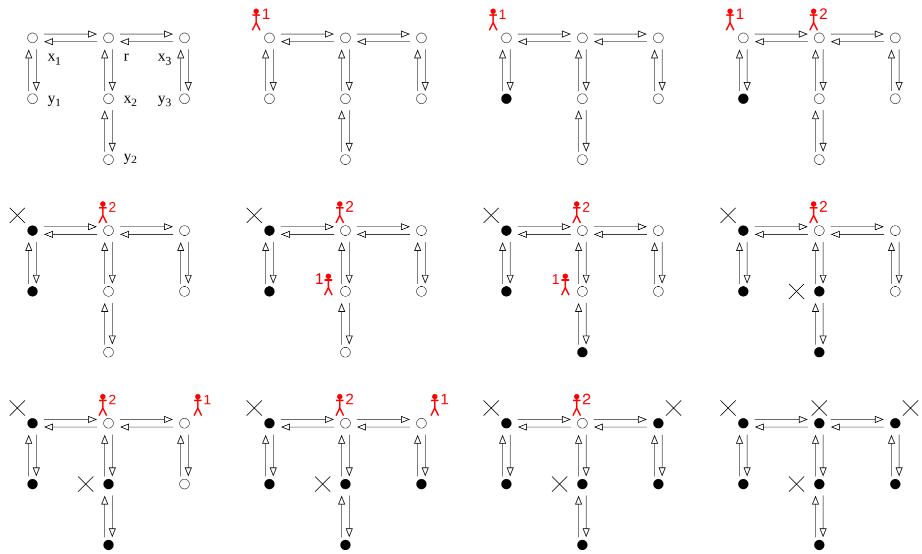
1. Introduction

In this paper, we study the *digraph processing* game, analogous to graph searching games [12]. This game aims at *processing*, or clearing, the vertices of a contaminated directed graph D. For this, we use a set of agents which are sequentially put and removed from the vertices of D. We are interested in two different measures and their tradeoffs: the minimum number of agents required to *clear* D and the minimum number of vertices that must be *covered* by an agent. The digraph processing game has been introduced in [6] for its relationship with the routing reconfiguration problem in Wavelength Division Multiplexing (WDM) networks. In this context, the goal is to reroute some connections that are established between pairs of nodes in a communication network, which can lead to interruptions of service. Each instance of this problem may be represented by a directed graph, called its *dependency digraph*, such that the reconfiguration problem is equivalent to the clearing of the dependency digraph. More precisely, the two measures presented above respectively correspond to the maximum number of simultaneous disruptions, and to the total number of requests disrupted during the rerouting of the connections. The equivalence between these two problems is detailed in Section 5.

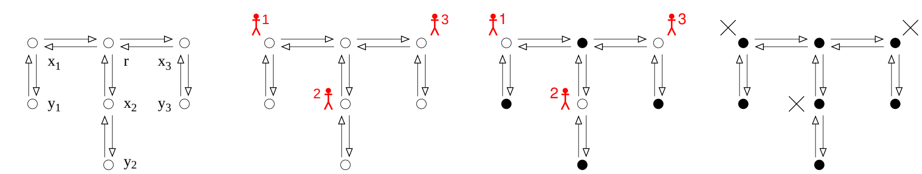
The digraph processing game is defined by the three following operations (or rules), which are very similar to the ones defining the *node search number* [2, 9, 12, 15, 17] of a graph, and whose goal is to *process*, or to clear, all the vertices of a digraph D.

- R_1 Put an agent at a vertex v of D;
- R_2 Remove an agent from a vertex v of D if all its outneighbors are either processed or occupied by an agent, and process v;
- R_3 Process an unoccupied vertex v of D if all its outneighbors are either processed or occupied by an agent.

A digraph whose vertices have all been processed is said *processed*. A sequence of such operations resulting in processing all vertices of D is called a *process strategy*. Note that, during a process strategy, an agent that has been removed from a (processed) vertex can be reused. The number of agents used by a strategy on a digraph D is the maximum number of agents present at the same time in D during the process strategy. A vertex is *covered* during a strategy if it is occupied by an agent at some step of the process strategy.



(a) A $(2,4)$-process strategy for D.



(b) A $(3,3)$-process strategy for D.

Figure 1: Different process strategies for a symmetric digraph D.

Figure 1 illustrates two process strategies for a symmetric digraph D of 7 vertices. The strategy depicted in Figure 1(a) first puts an agent at vertex x_1 (rule R_1), which let y_1 (rule R_3) be processed. A second agent is then put at r (rule R_1) allowing the vertex x_1 to be processed, and the agent on it to be removed (rule R_2). The procedure goes on iteratively, until all the vertices are processed. The depicted strategy uses 2 agents and covers 4 vertices. Another process strategy is depicted in Figure 1(b) that uses 3 agents and covers 3 vertices. Note that this latter strategy consists in putting agents at the vertices of a feedback vertex set[1] of minimum size.

[1]A set F of nodes of D is a feedback vertex set if the removal of all nodes in F makes D acyclic.

Clearly, to process a digraph D, it is sufficient to put an agent at every vertex of a feedback vertex set F of D (rule R_1), then the vertices of $V(D) \setminus F$ can be sequentially processed using rule R_3, and finally the vertices of F can be processed and all agents can be removed (rule R_2). In particular, a Directed Acyclic Graph (DAG) can be processed using 0 agent and thus covering no vertices. Indeed, to process a DAG, it is sufficient to process sequentially its vertices starting from the leaves (rule R_3). Note that any process strategy for a digraph D must cover all the vertices of a feedback vertex set of D (not necessarily simultaneously). Obviously, for any process strategy, the number of covered vertices is always at least the number of agents used.

The minimum number of agents required to process a digraph D (without constraint on the number of covered vertices) is called the *process number* [5–7], while the minimum number of covered vertices required to process D (without constraint on the number of agents) equals the size of a *minimum feedback vertex set* of D. In this paper, we are interested in tradeoffs between the minimum number of agents used by a process strategy and the minimum number of vertices it covers.

1.1. Definitions and Previous Results

Let D be a n-node directed graph. In the following, a (p,q)*-process strategy* for D denotes a process strategy for D using at most p agents and covering at most q vertices. When the number of covered vertices is not constrained, we write (p,∞)-process strategy. Similarly, when the number of agents is not constrained, we write (∞,q)-process strategy.

Process Number. The problem of finding the *process number* of a digraph D, was introduced in [6] as a metric of the routing reconfiguration problem (see Section 5). Formally,

Definition 1. The *process number* of D, denoted by $pn(D)$, is the smallest p such that there exists a (p,∞)-process strategy for D.

For instance, the digraph D of Figure 1 satisfies $pn(D) = 2$. Indeed, Figure 1(a) describes a process strategy using 2 agents, and it is easy to check that there is no process strategy using at most 1 agent. Digraphs whose process number is equal to 0 or 1 can easily be identified, as they respectively correspond to acyclic digraphs, and to graphs whose strongly connected components have a feedback vertex set of size at most 1 (which can be checked in linear time [7]). In [7] is also given an polynomial algorithm to recognize digraphs whose process number is equal to 2. However the problem of computing the process number of general digraphs is NP-complete and not in APX (i.e., admitting no polynomial-time approximation algorithm up to a constant factor, unless $P = NP$) [6]. A distributed polynomial-time algorithm to compute the process number of trees (or forests) with symmetric arcs has been proposed in [4]. Furthermore, a general heuristic to compute the process number of a digraph is described in [5]. In [19], Solano conjectured that computing the process number of a digraph can be solved, or approximated within a constant factor, in polynomial time if the set of covered vertices is given as part of the input. We disprove this conjecture, showing that computing the process number of a digraph remains not in APX (and so is NP-complete) in this situation (see Theorem 1).

When considering symmetric digraphs, which can be thought of as a directed version of an undirected graph, one notices that the process number is closely related to two other graph invariants, the *node search number* and the *pathwidth*. The node search number of a graph G, denoted by $sn(G)$, is the smallest p such that rules R_1 and R_2 (R_3 is omitted) are sufficient to process G using at most p agents. See [2, 9, 12, 15, 17] for more details. The pathwidth of a (undirected) graph G, denoted by $pw(G)$, was introduced by Robertson and Seymour in [18]. It has been proved in [10] by Ellis *et al.* that the pathwidth and the node search number are equivalent, that is for any graph G, $pw(G) = sn(G) - 1$. The relationship between these parameters and the process number has been described in [6]: $pw(G) \leq pn(G) \leq pw(G) + 1$ (and so $sn(G) - 1 \leq pn(G) \leq sn(G)$), where $pn(G)$ is the process number of the digraph built from G by replacing each edge by two opposite arcs. Since computing the pathwidth of a graph is NP-complete [16] and not in APX [8], determining these parameters is as hard.

Minimum Feedback Vertex Set. Given a digraph D, the problem of finding a process strategy that minimizes the number of nodes covered by agents is equivalent to the one of computing a *minimum feedback vertex set* (MFVS) of D. Computing such a set is well known to be NP-complete and APX-hard [14]. A 2-approximation algorithm is known in undirected graphs [1] and in symmetric digraph (where a feedback vertex set is a vertex cover of the underlying graph). As far as we know, the best approximation algorithm for computing a MFVS in general n-node digraphs has ratio $\log n \log\log n$ [11].

We define below the parameter $mfvs(D)$, using the notion of (p,q)-process strategy, corresponding to the size of a MFVS of D.

Definition 2. Let $mfvs(D)$ denote the smallest q such that there exists a (∞, q)-process strategy for D.

As an example, the digraph D of Figure 1 satisfies $mfvs(D) = 3$. Indeed for $i \in \{1,2,3\}$, it is easy to see that either x_i or y_i must be in any feedback vertex set (FVS) of D because of the cycle (x_i, y_i, x_i). Furthermore the removal of x_1, x_2, and x_3 from D is sufficient to break all the cycles. Thus these three nodes form a MFVS of D, and so $mfvs(D) = 3$. The corresponding strategy, covering $mfvs(D) = 3$ nodes by agents, is described in Figure 1(b).

As mentioned above, $mfvs(D) \geq pn(D)$. Moreover, the gap between these two parameters may be arbitrarily large. For example consider a symmetric path P_n composed of $n \geq 4$ nodes $u_1, u_2, \ldots, u_n$ with symmetric arcs between u_i and u_{i+1} for $i = 1, \ldots, n-1$. We get $mfvs(P_n) = \lfloor \frac{n}{2} \rfloor$ while $pn(P_n) = 2$. Indeed either u_i or u_{i+1} must be in any FVS of P_n, and so we deduce that nodes $u_2, u_4, u_6, \ldots$ form a MFVS of P_n. Furthermore $pn(P_n) \geq 2$ because P_n is strongly connected and $mfvs(P_n) > 1$. We then describe a process strategy for P_n using 2 agents: we put the first agent at u_1 (R_1), we put the second agent at u_2 (R_1), we process u_1 removing the agent from it (R_2), we put this agent at u_3 (R_1), we process u_2 removing the agent from it (R_2), we put an agent at u_4 (R_1), and so on.

Remark that this process strategy for P_n uses the optimal number of agents, $pn(D) = 2$, but all the n nodes are covered by an agent at some step of the process strategy. For this digraph P_n, it is possible to describe a $(pn(D) = 2, mfvs(D) = \lfloor \frac{n}{2} \rfloor)$-process strategy, that is a process strategy for P_n minimizing both the number of agents and the total number of covered nodes. We put the first agent at u_2 (R_1), we process u_1 (R_3), we put the second agent at u_4 (R_1), we process u_3 (R_3), we process u_2 removing the agent from it (R_2), we put this agent at u_6 (R_1), and so on. Unfortunately such good tradeoffs are not always possible (it is the case for the digraph of Figure 1 as explained later). Actually, we prove in this paper that there exist some digraphs for which minimizing one of these objectives arbitrarily impairs the quality of the solution for the other one. In the following, we define formally the tradeoff metrics we will now study.

Tradeoff Metrics. We introduce new tradeoff metrics in order to study the loss one may expect on one parameter when adding a constraint on the other. In particular, what is the minimum number of vertices that must be covered by a process strategy for D using $pn(D)$ agents ? Similarly, what is the minimum number of agents that must be used to process D while covering $mfvs(D)$ vertices ?

Definition 3. Given an integer $q \geq mfvs(D)$, we denote by $pn_q(D)$ the minimum p such that a (p,q)-process strategy for D exists. We write $pn_{mfvs+r}(D)$ instead of $pn_{mfvs(D)+r}(D)$, $r \geq 0$.

Definition 4. Given an integer $p \geq pn(D)$, we denote by $mfvs_p(D)$ the minimum q such that a (p,q)-process strategy for D exists. We write $mfvs_{pn+r}(D)$ instead of $mfvs_{pn(D)+r}(D)$, $r \geq 0$.

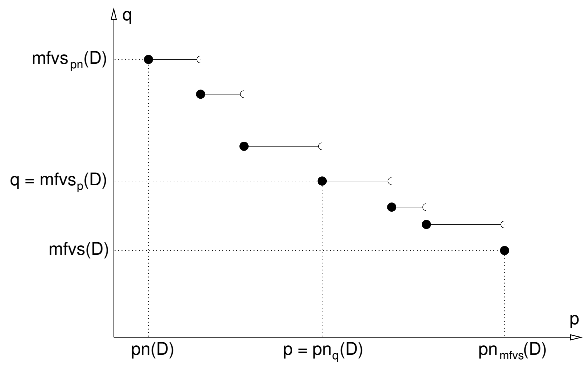


Figure 2: $mfvs_p(D)$ function of p for a digraph D. Filled circles represent minimal values of D.

Intuitively $pn_{mfvs}(D)$ is the minimum number of agents required by a process strategy minimizing the number of covered vertices, and $mfvs_{pn}(D)$ is the minimum number of vertices that must be covered by a process strategy using the minimum number of agents. Note that, $pn_{mfvs}(D)$ is upper bounded by the maximum MFVS of the strongly connected components of D. Another straightforward remark is that $mfvs_{mfvs}(D) = mfvs(D)$ for any digraph D.

To illustrate the pertinence of these tradeoff metrics, consider the digraph D of Figure 1. Recall that $pn(D) = 2$ and $mfvs(D) = 3$. We can easily verify that there does not exist a $(2,3)$-process strategy for D, that is a process strategy minimizing both p and q. On the other hand, we can exhibit a $(2,4)$-process strategy (Figure 1(a)) and a $(3,3)$-process strategy (Figure 1(b)) for D. Hence, we have: $pn_{mfvs}(D) = 3$ while $pn(D) = 2$, and $mfvs_{pn}(D) = 4$ while $mfvs(D) = 3$. Intuitively for these two process strategies, we can not decrease the value of one parameter without increasing the other.

We generalize this concept through the notion of *minimal values* of a digraph D. We say that (p,q) is a minimal value of D if $p = pn_q(D)$ and $q = mfvs_p(D)$. Note that $(pn(D), mfvs_{pn}(D))$ and $(pn_{mfvs}(D), mfvs(D))$ are both minimal values by definition (and may be the same). For the digraph of Figure 1, there are two minimal values: $(2,4)$ and $(3,3)$. Figure 2 depicts the variations of the minimum number q of vertices covered by a p-strategy for a digraph D ($p \geq pn(D)$), i.e., $mfvs_p(D)$ as a function of p. Clearly, it is a non-increasing function upper bounded by $mfvs_{pn}(D)$ and lower bounded by $mfvs(D)$.

Filled circles of Figure 2 represent the shape of minimal values of D. Clearly for a given digraph D, the number of minimal values is at most linear in the number of nodes. We now give an example of a family of n-node digraph for which the number of minimal value is $\Omega(\sqrt{n})$. Intuitively, it means that, in those digraphs D, starting from the optimal number of agents $pn(D)$, each extra agent added allows to strictly decrease the number of covered vertices, until the optimal, $mfvs(D)$, is reached. Let H_n be the symmetric directed star with $n \geq 3$ branches of length 2 (for instance, H_3 is the digraph of Figure 1), and let G_k be the graph that consists of the disjoint union of $H_3, \cdots, H_k$, $k \geq 3$. Then, for any $0 \leq i \leq k-2$, $(pn(G_k)+i, mfvs(G_k)+k-2-i) = (2+i, (k(k+1)/2)-5+k-i)$ are minimal values (this can be easily proved using the easy results described in Section 2.1).

1.2. Our Results

Our results consist in an analysis of the behaviour of the two given tradeoff measures both in general digraphs and in symmetric digraphs. As mentioned above, in general, no process strategy minimizes both the number of agents and the number of covered vertices (see example in Figure 1). Hence, we are interested in the loss on one measure when the other is constrained. In particular, we are interested in the ratios $\frac{pn_{mfvs}(D)}{pn(D)}$ and $\frac{mfvs_{pn}(D)}{mfvs(D)}$. This study involves various theorems on the complexity of estimating this loss (Section 2) and the existence of digraphs for which it can be arbitrarily large (Section 3). We also study in Section 4 the case of symmetric digraphs. Finally we describe in Section 5 the relation between the routing reconfiguration problem and the processing game.

More precisely, we begin by disproving a conjecture from Solano [19] (Theorem 1). Then, we prove that for all $\alpha, \beta \geq 0$, the problems of determining the parameters $\alpha.pn_{mfvs}(D) + \beta.pn(D)$ and $\alpha.mfvs_{pn}(D) + \beta.mfvs(D)$ are NP-complete (Theorem 2). In particular, the problem of determining $pn_{mfvs}(D)$ is not in APX and the problem of determining $mfvs_{pn}(D)$ is APX-hard (Theorem 2). Then, we prove that for any $q \geq 0$ (resp. for any $p \geq 0$), the ratio $\frac{pn_{mfvs+q}(D)}{pn(D)}$ (resp. $\frac{mfvs_{pn+p}(D)}{mfvs(D)}$) is not bounded even in the class of bounded process number digraphs (Theorem 3 and Theorem 4). However we prove that $\frac{mfvs_{pn}(D)}{mfvs(D)} \leq pn(D)$ for any symmetric digraph D (Lemma 5).

In Section 5, we detail the relationship between the processing game and the reconfiguration routing problem. In this context, any instance of the routing reconfiguration problem may be represented by a directed graph, called the dependency digraph of this instance, such that the routing reconfiguration problem is equivalent to the processing of this digraph. We prove the reverse, that is, any digraph is the dependency digraph of an instance of the reconfiguration problem (Theorem 7).

2. Complexity Results

This section is devoted to the study of the complexity of the problems related to the parameters introduced in Section 1.1. First, we need to define some digraphs.

2.1. Definition of some useful digraphs.

Let H_n be a symmetric directed star with $n \geq 3$ branches each of which contains two vertices (the root r being at distance 2 from any leaf), with a total of $2n+1$ vertices. H_3 is represented in Figure 1. It is easy to check that $pn(H_n) = 2$. Indeed 1 agent is obviously not sufficient and there exists a $(2, n+1)$-process strategy for H_n: an agent is put at the central node r, then we successively put an agent at a vertex x adjacent to r, the remaining neighbor of x (different from r) is processed, and we process x itself relieving the agent on it. Then, the same process is applied until all vertices adjacent to r are processed, and finally we process r. Figure 1(a) represents a $(2,4)$-process strategy for H_3. Moreover, the single MFVS of H_n is the set X of the n vertices adjacent to r. It is easy to check that the single process strategy occupying only the vertices of X consists in putting n agents at all vertices of X. No agent can be removed while all agents have not been put. Thus this strategy is a (n,n)-process strategy, and $pn_{mfvs}(H_n) = n$. See Figure 1(b) for such a process strategy for H_3. To summarize, the two minimal values of H_n are $(pn(H_n), mfvs_{pn}(H_n)) = (2, n+1)$ and $(pn_{mfvs}(H_n), mfvs(H_n)) = (n,n))$.

Let K_n be a symmetric complete digraph of n nodes. It is easy to check that the unique minimal value of K_n is $(pn(K_n), mfvs(K_n)) = (n-1, n-1)$.

Let $D = (V,A)$ be a symmetric digraph with $V = \{u_1, \ldots, u_n\}$. Let $\hat{D} = (V', A')$ be the symmetric digraph where $V' = V \cup \{v_1, \ldots, v_n\}$, and $\hat{D}$ is obtained from D by adding two symmetric arcs between u_i and v_i for $i = 1, \ldots, n$. It is easy to show that there exists an optimal process strategy for $\hat{D}$ such that the set of occupied vertices is V. Indeed, note that, for all i, at least one of u_i or v_i must be covered by an agent (any FVS of D contains at least one of v_i or u_i). Furthermore if some step of a process strategy for $\hat{D}$ consists in putting an agent at some vertex v_i, then the process strategy can be easily transformed by putting an agent at u_i instead. In particular, $mfvs_{pn}(\hat{D}) = n$.

2.2. NP-completeness.

Before proving that computing the tradeoff parameters introduced in Section 1.1 are NP-complete, we disprove a conjecture of Solano about the complexity of computing the process number of a digraph D.

Indeed a possible approach for computing the process number, proposed by Solano in [19], consists of the following two phases: 1) finding the subset of vertices of the digraph at which an agent will be put, and 2) deciding the order in which the agents will be put at these vertices. Solano conjectures that the complexity of the process number problem resides in Phase 1 and that Phase 2 can be solved, or approximated within a constant factor, in polynomial time [19]. We disprove this conjecture :

Theorem 1. *Computing the process number of a digraph is not in APX (and thus NP-complete), even when the subset of vertices of the digraph at which an agent will be put is given.*

Proof. Let D be any symmetric digraph. Let us consider the problem of computing an optimal process strategy for $\hat{D}$ when the set of vertices covered by agents is constrained to be V. By the remark in Section 2.1, such an optimal strategy always exists. It is easy to check that this problem is equivalent to the one of computing the node search number (and so the pathwidth) of the underlying undirected graph of D which is NP-complete [16] and not in APX [8]. □

Theorem 2. *Let* $\alpha, \beta \geq 0$ *be fixed, with* $\max\{\alpha, \beta\} > 0$*. The problem that takes a digraph D as an input and that aims at determining:*

- $\alpha.pn_{mfvs}(D) + \beta.pn(D)$ *is not in APX,*
- $\alpha.mfvs_{pn}(D) + \beta.mfvs(D)$ *is APX-hard.*

Proof. The two cases for $\alpha = 0$ and $\beta > 0$ clearly holds from the literature. Now, let us assume $\alpha > 0$.

- We start with $\alpha.pn_{mfvs}(D) + \beta.pn(D)$.

  Let us first consider the case $\beta = 0$. That is, let us show that the problem of determining pn_{mfvs} is not in APX. Indeed, let $\mathcal{D}$ be the class of all digraphs $\hat{D}$ obtained from some symmetric digraph D. For any symmetric digraph D, the problem of computing $pw(D)$ (where $pw(D)$ is the pathwidth of the underlying graph of the symmetric digraph of D) is not in APX, and $pn(\hat{D}) = pn_{mfvs}(\hat{D}) = pw(D) + 1$ (see Theorem 1). Hence, the problem of determining pn_{mfvs} is not in APX.

  Assume now that $\beta > 0$. To prove that determining $\alpha.pn_{mfvs}(D) + \beta.pn(D)$ is not in APX, let D_1 be the disjoint union of H_n and any n-node digraph D. First, let us note that $pn_{mfvs}(D_1) = pn_{mfvs}(H_n)$ because $pn_{mfvs}(D) \leq n - 1$ and $pn_{mfvs}(H_n) = n$. Since $pn(D_1) = \max\{pn(D), pn(H_n)\}$ and $pn(H_n) = 2$, we get that $\alpha.pn_{mfvs}(D_1) + \beta.pn(D_1) = \alpha.n + \beta \max\{pn(D), 2\}$. So, the NP-completeness comes from the NP-completeness of the process number problem.

- We now consider $\alpha.mfvs_{pn}(D) + \beta.mfvs(D)$.

  When $\beta = 0$, let us prove that the problem of determining $mfvs_{pn}$ is APX-hard. Let D_2 be the disjoint union of K_n and any n-node digraph D. First let us note that $pn(D_2) = \max\{pn(K_n), pn(D)\}$ because the process number of any digraph is the maximum for the process numbers of its strongly connected components. It is easy to show that $pn(D_2) = pn(K_n) = n-1$ because $pn(D) \leq n-1$. Hence, when D must be processed, $n-1$ agents are available. So, in order to minimize the number of nodes covered by agents, the agents must be placed on a MFVS of D. Thus $mfvs_{pn}(D_2) = n-1+mfvs(D)$, and the result follows because computing $mfvs(D)$ is APX-hard.

Assume now that $\beta > 0$. To prove that determining $\alpha.mfvs_{pn}(D) + \beta.mfvs(D)$ is APX-hard, let D_3 be the disjoint union of K_n, H_n, and D. Again, $pn(D_3) = \max\{pn(K_n), pn(H_n), pn(D)\}$. It is easy to show that $pn(D_3) = pn(K_n) = n-1$ because $pn(H_n) = 2$ and $pn(D) \leq n-1$. Moreover, any process strategy of D_3 using $n-1$ agents must cover $n-1$ nodes of K_n, $n+1$ nodes of H_n ($mfvs(H_n) = n$ but one extra agent is needed to cover only n nodes), and $mfvs(D)$ nodes of D (because $n-1$ agents are available and $mfvs(D) \leq n-1$). Hence, $mfvs_{pn}(D_3) = (n-1)+(n+1)+mfvs(D)$. Furthermore $mfvs(D_3) = (n-1)+n+mfvs(D)$ because $mfvs(K_n) = n-1$ and $mfvs(H_n) = n$. Thus $\alpha.mfvs_{pn}(D_3) + \beta.mfvs(D_3) = (\alpha+\beta)(mfvs(D)+2n) - \beta$. The result follows the APX-hardness of the MFVS problem.

□

Corollary 1. For an input digraph D and two integers $p \geq 0$ and $q \geq 0$, and any $\alpha, \beta \geq 0$ ($\{\alpha, \beta\} \neq \{0,0\}$) the problems of determining:

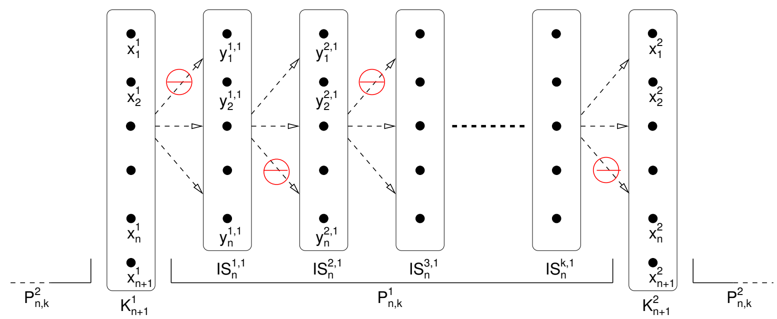
- $\alpha.pn_{mfvs+q}(D) + \beta.pn(D)$ are not in APX,
- $\alpha.mfvs_{pn+p}(D) + \beta.mfvs(D)$ are APX-hard.

3. Behaviour of ratios in general digraphs

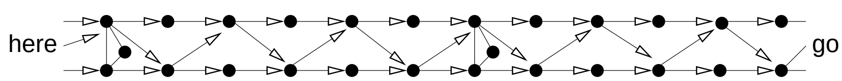
In this section, we study the behaviours of parameters introduced in Section 1.1 and their ratios, showing that, in general, good tradeoffs are impossible.

Theorem 3. *For any $C > 0$ and any integer $q \geq 0$, there exists a digraph D such that $\frac{pn_{mfvs+q}(D)}{pn(D)} > C$.*

Proof. Consider the symmetric directed star H_n defined in Section 2.1. Let now D be the digraph consisting of $q+1$ pairwise disjoint copies of H_n. So D has $q+1$ strongly connected components. We get $mfvs(D) = (q+1)n$. By definition, any $(pn_{mvfs+q}(D), mfvs(D)+q)$-process strategy for D covers at most $q(n+1)+n$



(a) $D_{n,k}$ of Theorem 4 and Corollary 3 (Case k odd). The red symbol $\ominus$ represents the inexistence of arcs between these subgraphs. The arcs from $V(D_{n,k}) \setminus V(K^1_{n+1})$ to $V(K^1_{n+1})$ are not represented.



(b) $D_{2,5}$ in Cor. 3 where the arcs from all vertices to one triangle K^1_3 have been omitted.

Figure 3: Digraph $D_{n,k}$ described in Theorem 4 and Corollary 3.

nodes. Therefore, there exists at least one of the $q+1$ strongly connected components for which at most n nodes must be covered. Thus to process this component, n agents are required. Indeed (n,n) is a minimal value of H_n, and by definition we cannot decrease the first value without increasing the second one. Hence, $pn_{mfvs+q}(D) = n$ while $pn(D) = 2$. Taking $n > 2C$, we get $\frac{pn_{mfvs+q}(D)}{pn(D)} > C$. $\square$

Note that if it is allowed to cover $mfvs(D)+q+1$ nodes during the process strategy (instead of $mfvs(D)+q$), then the number of agents required is $pn(D)$. In other words, for the digraph D described in the proof of Theorem 3, we get $\frac{pn_{mfvs+q+1}(D)}{pn(D)} = 1$ while $\frac{pn_{mfvs+q}(D)}{pn(D)} = \frac{n}{2}$.

Corollary 2. For any $C > 0$, there exists a digraph D such that $\frac{pn_{mfvs}(D)}{pn(D)} > C$.

In the sequel, we present similar results for the second ratio.

Theorem 4. *For any $C > 0$ and any integer $p \geq 0$, there exists a digraph D such that $\frac{mfvs_{pn+p}(D)}{mfvs(D)} > C$.*

Proof. Let $n \geq 2$ and let $k \geq 1$ be an odd integer. Let us consider the digraph $D_{n,k}$ built as follows. Let $IS^1_n, \cdots, IS^k_n$ be k independent sets, each IS^t_n $(1 \leq t \leq k)$ having

n vertices: $y_1^t, y_2^t, \ldots, y_n^t$. Let $P_{n,k}$ be the digraph obtained from the k independent sets IS_n^t ($1 \leq t \leq k$) by adding the arcs from y_i^t to y_j^{t+1}, for $1 \leq j \leq i \leq n$ and $t = 1, 3, \ldots, k-2$, and from y_i^t to y_j^{t+1}, for $1 \leq i \leq j \leq n$ and $t = 2, 4, \ldots, k-1$. Let K_{n+1} be the symmetric clique with $n+1$ nodes: $x_1, x_2, \ldots, x_{n+1}$.

The digraph $D_{n,k}$ is obtained from two copies $P_{n,k}^1, P_{n,k}^2$ of $P_{n,k}$ and two copies K_{n+1}^1, K_{n+1}^2 of K_{n+1}, by adding the following arcs. In what follows, $y_j^{t,a}$ denotes the j^{th} vertex in the t^{th} independent set of $P_{n,k}^a$, where $j \leq n$, $t \in \{1, k\}$, $a \in \{1, 2\}$, and x_j^a denotes the j^{th} vertex of K_n^a, where $j \leq n+1$, $a \in \{1, 2\}$. There are arcs from x_i^a to $y_j^{1,a}$, for $1 \leq i \leq j \leq n$ and $a = 1, 2$, and from $y_i^{k,a}$ to x_j^b, for $1 \leq i \leq j \leq n$, $a = 1, 2$ and $b = 3 - a$. Finally there is an arc from each node of $V(D_{n,k}) \setminus V(K_{n+1}^1)$ to each node of $V(K_{n+1}^1)$. Note that these last arcs are not needed to obtain the results but help make the proof less technical.

Figure 3(a) shows the general shape of $D_{n,k}$, where the red symbol $\ominus$ represents the inexistence of arcs between these subgraphs. $D_{2,5}$ is depicted in Figure 3(b). For not overloading the figures, the arcs from $V(D_{n,k}) \setminus V(K_{n+1}^1)$ to $V(K_{n+1}^1)$ are not represented.

Clearly, $mfvs(D_{n,k}) = 2n$, and any MFVS consists of $\{x_1^1, \ldots, x_n^1\}$ plus n vertices of K_{n+1}^2.

First, note that to process one vertex of K_{n+1}^1, there must be a step of any process strategy for $D_{n,k}$ where n agents are simultaneously occupying n nodes of K_{n+1}^1. Hence, $pn(D_{n,k}) \geq n$. Note that, similarly, any process strategy for $D_{n,k}$ must occupy n vertices of K_{n+1}^2. Moreover, because of the arcs from $V(D_{n,k}) \setminus V(K_{n+1}^1)$ to $V(K_{n+1}^1)$, any agent that is placed at some vertex in $V(D_{n,k}) \setminus V(K_{n+1}^1)$ can only be removed when all vertices of K_{n+1}^1 are occupied or processed. Consider any process strategy S for $D_{n,k}$ (in particular, S uses at least n agents) and let s_0 be the first step of S that does not consist in placing an agent at some vertex of K_{n+1}^1. By above remark, after step $s_0 - 1$ of S, n agents are occupying n vertices of $V(K_{n+1}^1)$. Up to reorder the first $s_0 - 1$ steps of S, we obtain a process strategy for $D_{n,k}$ that starts by placing n agents at n vertices of $V(K_{n+1}^1)$, without increasing the number of agents used nor the number of vertices occupied by S. Moreover, if the vertex of $V(K_{n+1}^1)$ that is not occupied is x_i^1 with $i < n+1$, it means that an agent is placed at x_{n+1}^1 during the first n steps of the strategy. Replacing this operation by the placement of an agent at x_i^1 instead of x_{n+1}^1 does not modify the remaining part of the strategy (but the operation "remove the agent from x_{n+1}^1" which is replaced by "remove the agent from x_i^1") since the vertex x_{n+1}^1 can be processed immediatly when the n other vertices of K_{n+1}^1 are occupied. Hence, we may assume that S starts by placing agents at $\{x_1^1, \ldots, x_n^1\}$ and then processes x_{n+1}^1.

Second, any process strategy for any graph can easily be modified, without increasing (possibly decreasing) the number of used agents nor the number of occupied vertices, in such a way that the strategy processes all possible vertices before placing or removing agents. In other words, the rule R_3 can be made prioritary without increasing the considered parameters. Therefore, any process strategy S for $D_{n,k}$ can be modified , without increasing the number of agents used nor the number of vertices occupied by S, into a strategy that first places n agents at $\{x_1^1,\dots,x_n^1\}$, then processes x_{n+1}^1 and all vertices of $P_{n,k}^2$, and finally that mimicks S. Such a strategy is called a *good* process strategy for $D_{n,k}$.

Third, $pn(D_{n,k}) \leq n+1$ as proved by the following strategy S^*. First, place n agents at $\{x_1^1,\dots,x_n^1\}$, then process all vertices of $P_{n,k}^2$ and then x_{n+1}^1. In the next sentence, $y_i^{0,1}$ denotes x_i^1 and $y_i^{k+1,1}$ denotes x_i^2, $i \leq n$. Then, for $j = 1\dots k+1$, the j^{th} phase of S^* consists of the following: for $i = 1\dots n$, place an agent at $y_{n-i+1}^{j,1}$ if j odd (resp., at $y_i^{j,1}$ if j even) and remove the agent at $y_{n-i+1}^{j-1,1}$ (resp., at $y_i^{j-1,1}$ if j even). Finally, process all vertices of K_{n+1}^2.

Let p, $0 \leq p \leq n-2$ (we choose $n \geq p-2$). Let S be a good process strategy for $D_{n,k}$ that uses $n+1+p$ agents (which exists by the previous remarks). We assume that S minimizes the number q of independent sets $IS_n^{t,1}$ of $D_{n,k}$ for which a vertex is occupied during the execution of S. Such an independent set is said *touched*. Note that the transformation that makes a strategy *good* does not increase the number of touched independent sets. Therefore, $2n+q \leq mfvs_{n+1+p}(D_{n,k})$ since any strategy occupies n vertices in each clique plus at least one vertex per touched independent set. In the sequel, we will prove that $q \geq k$, i.e., all independent sets of $P_{n,k}^1$ must be touched, and then, taking $k > 2n(C-1)$, we get that $\frac{mfvs_{pn+p}(D_{n,k})}{mfvs(D_{n,k})} = \frac{mfvs_{pn+p}(D_{n,k})}{2n} \geq \frac{mfvs_{n+1+p}(D_{n,k})}{2n} \geq \frac{2n+k}{2n} > C$.

It remains to prove that S touches all the k independent sets of $P_{n,k}^1$. To do so, we will modify S, possibly increasing the number of occupied vertices but without increasing the number of touched independent sets.

Since S is good, it first places n agents at $\{x_1^1,\dots,x_n^1\}$, then processes x_{n+1}^1 and all vertices of $P_{n,k}^2$. We set $x_i^1 = y_i^{0,1}$, for all $i \leq n$. Let $S = S^0$. Let $0 \leq j < k$ and let S^j be the strategy that mimicks the j first phases of S^* and then performs in the same order those movements of S^0 that concern the unprocessed vertices at this step. We prove by induction on $j < k$ that S^j can be transformed into the good process strategy S^{j+1} for $D_{n,k}$ satisfying the desired properties without increasing the number of touched independent sets. Clearly, S^0 is a good process strategy for $D_{n,k}$ that satisfies these properties.

Assume that, for some $0 \leq j < k-1$, S^j is a good process strategy that satisfies the desired properties. Then, S^j starts by occupying the vertices of $\{x_1^1,\dots,x_n^1\}$, processes x^1_{n+1} and the vertices of $P^2_{n,k}$ and then occupies and processes successively all vertices of $IS^{r,1}_n$, $r = 1 \dots j$ until all vertices of $IS^{j,1}_n$ are occupied. Let s_j be the step of S^j when it occurs. We first prove that S^j touches $IS^{j+1,1}_n$. Indeed, if j is even, there are n vertex-disjoint paths from $y^{j+1,1}_n$ (resp., from $y^{j+1,1}_1$ if j is odd) to $x^2_1, \dots, x^2_n$. While $y^{j+1,1}_n$ (resp., from $y^{j+1,1}_1$ if j is odd) is not processed, no agent in $IS^{j,1}_n$ can be removed, and thus only $p+1 \leq n-1$ agents are available. Therefore, the only way to process $y^{j+1,1}_n$ (resp., from $y^{j+1,1}_1$ if j is odd) is to place an agent at it. Hence, there is a step of S^j (hence, of S^0) that consists of placing an agent at $y^{j+1,1}_n$ (resp., $y^{j+1,1}_1$ if j is odd). Hence, S^0 touches $IS^{j+1,1}_n$. To conclude, we modify S^j by adding after step s_j the $j+1^{th}$ phase of S^*. That is, after step j, the strategy successively occupies the vertices of $IS^{j+1,1}_n$ removing the agents at $IS^{j,1}_n$ until all vertices of $IS^{j+1,1}_n$ are occupied and all vertices of $IS^{j,1}_n$ have been processed. Then, the strategy mimicks the remaining steps of S^j. The strategy obtained in such a way is clearly S^{j+1} that satisfies all desired properties. In particular, the obtained strategy is a good process strategy for $D_{n,k}$ that touches the same independent sets as S^0. □

Note that there exists a $(pn(D)+p+1, mfvs(D))$-process strategy for the digraph $D_{n,k}$ described in the proof of Theorem 4 whereas the minimum q such that a $(pn(D)+p, q)$-process strategy for $D_{n,k}$ exists, is arbitrarily large.

Corollary 3. For any $C > 0$, there exists a digraph D such that $\frac{mfvs_{pn}(D)}{mfvs(D)} > C$.

We obtain this result by considering the digraph $D_{n,k}$ described in Figure 3(a), with $n = 2$ and $k \geq 1$ (Figure 3(b) represents $D_{2,5}$). This digraph is such that $pn(D_{2,k}) = 3$ and $mfvs(D_{2,k}) = 4$ while $\frac{mfvs_{pn}(D_{2,k})}{mfvs(D_{2,k})} = \frac{k+4}{4}$ is unbounded.

Lemma 5 in Section 4 shows that, in the class of symmetric digraphs with bounded process number, $\frac{mfvs_{pn}(D)}{mfvs(D)}$ is bounded.

4. Behaviour of ratios in symmetric digraphs

We address in this section the behaviour of $\frac{mfvs_{pn}(D)}{mfvs(D)}$ for symmetric digraphs D. Note that the behaviours of $\frac{pn_{mfvs+q}(D)}{pn(D)}$ and $\frac{pn_{mfvs}(D)}{pn(D)}$ have already been studied in Section 3 for symmetric digraphs with bounded process number.

Lemma 5. *For any symmetric digraph D, $\frac{mfvs_{pn}(D)}{mfvs(D)} \leq pn(D)$.*

Proof. Without loss of generality, we prove the lemma for a connected digraph D. Let S be a $(pn(D), mfvs_{pn}(D))$-process strategy for $D = (V,E)$. Let $O \subseteq V$ be the

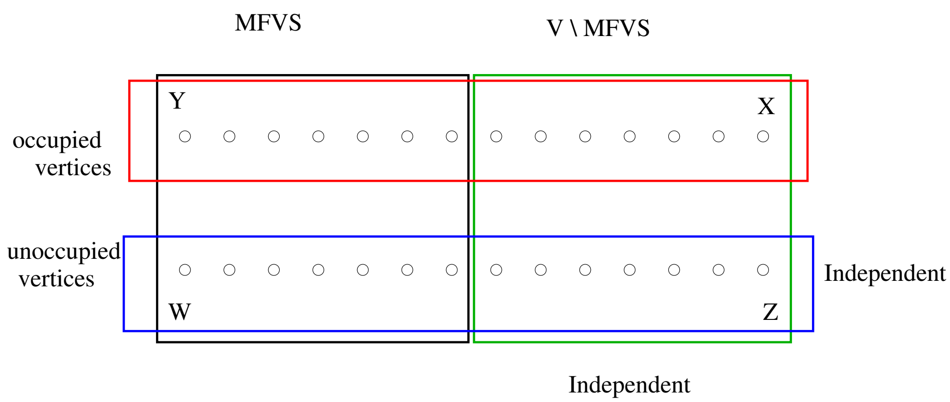


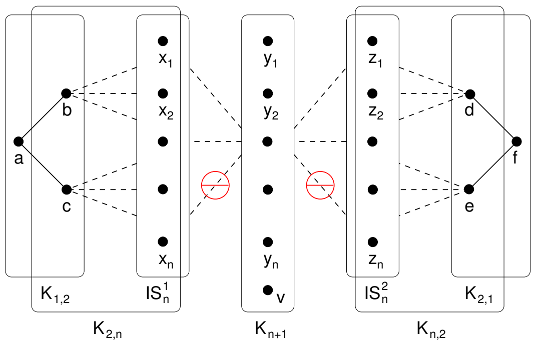
Figure 4: Proof of Lemma 5

set of vertices occupied by an agent during the execution of S. Let F be a MFVS of D. Let us partition V into $(Y,X,W,Z) = (O \cap F, O \setminus F, F \setminus O, V \setminus (O \cup F))$. Since D is symmetric, $V \setminus F$ is an independent set because it is the complementary of a MFVS. Since the vertices not occupied by S have all their neighbors occupied, $V \setminus O$ is an independent set. Given $V' \subseteq V$, $N(V')$ denotes the set of neighbors of the vertices in V'. The partition is illustrated in Figure 4.

First, note that $|N(W) \cap X| \leq pn(D)|W|$, because, for any vertex $v \in W$ to be processed, all its neighbors must be occupied by an agent. Thus, the maximum degree of v is $pn(D)$.

Then, we prove that $|X \setminus N(W)| \leq (pn(D)-1)|Y|$. Let $R = X \setminus N(W)$. Because $X \cup Z$ is an independent set, for any $v \in R$, $N(v) \subseteq Y$. Let $T = N(R) \subseteq Y$. Note that $N(T) \cap R = R$ because D is connected and symmetric. Let us order the vertices of $T = \{v_1, \cdots, v_t\}$ in the sequence in which they are processed (when the agents are removed) when executing S. For any i, $1 \leq i \leq t$, let $N_i = \bigcup_{j \leq i} N(v_j) \cap R$. We aim at proving that $|N_1| < pn(D)$ and $|N_{i+1} \setminus N_i| < pn(D)$ for any $i < t$. Hence, we obtain $|N_t| = |R| \leq (pn(D)-1)|T| \leq (pn(D)-1)|Y|$.

Let us consider the step of S just before an agent is removed from v_1. Let $v \in N_1 \neq \emptyset$. Since the agent will be removed from v_1, either v has already been processed or is occupied by an agent. We prove that there is a vertex in $N(v) \subseteq T$ that has not been occupied yet and thus v must be occupied. Indeed, otherwise, all neighbors of v are occupied (since, at this step, no agents have been removed from the vertices of T) and the strategy can process v without placing any agent on v, contradicting the fact that S occupies the fewest vertices as possible. Therefore, just before an agent is to be removed from v_1, all vertices of N_1 are occupied by an



(a) SD_n

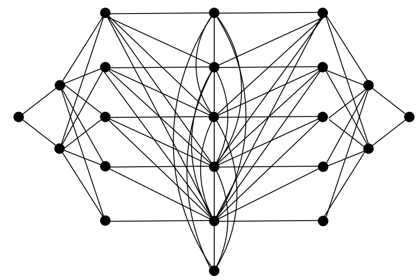

(b) SD_5

Figure 5: Symmetric digraph SD_n of Lemma 6 (Figure 5(a)) and instance of SD_n when $n = 5$ (Figure 5(b)). The red symbol ⊖ represents the absence of arcs.

agent. Hence, $|N_1| < pn(D)$.

Now, let $1 < i \leq t$. Let us consider the step of S just before an agent is removed from v_i. Let $v \in N_i \setminus N_{i-1}$ if such a vertex exists. Since the agent will be removed from v_i, either v has already been processed or is occupied by an agent. We prove that there is a vertex in $N(v) \subseteq T \setminus N_{i-1}$ that has not been occupied yet and thus v must be occupied. Indeed, otherwise, all neighbors of v are occupied (since, at this step, no agents have been removed from the vertices of $T \setminus N_{i-1}$) and the strategy can process v without placing any agent on v, contradicting the fact that S occupies the fewest vertices as possible. Therefore, just before an agent to be removed from v_i, all vertices of $N_{i+1} \setminus N_i$ are occupied by an agent. Hence, $|N_{i+1} \setminus N_i| < pn(D)$.

To conclude: $mfvs_{pn}(D) = |O| = |Y| + |X|$ and $X = |X \setminus N(W)| + |N(W) \cap X|$. Hence, $mfvs_{pn}(D) \leq pn(D)(|Y| + |W|) = pn(D)|F| = pn(D).mfvs(D)$. □

Lemma 6. *For any given $\varepsilon > 0$, there exists a symmetric digraph D such that* $3 - \varepsilon \leq \frac{mfvs_{pn}(D)}{mfvs(D)} < 3$.

Proof. Let $n \geq 1$. Let us consider the digraph SD_n built as follows. Let IS_n^1 and IS_n^2 be two independent sets of n nodes each: respectively $x_1, \ldots, x_n$ and $z_1, \ldots, z_n$. Let K_{n+1} be a symmetric clique of $n+1$ nodes $y_1, \ldots, y_n, y_{n+1} = v$. The digraph SD_n is built starting from the disjoint union of IS_n^1, IS_n^2, K_{n+1} and 6 isolated vertices $\{a, b, c, d, e, f\}$ by adding the following arcs. There are symmetric arcs between the nodes x_i and y_j and the nodes z_i and y_j, for any $1 \leq i \leq j \leq n$. Furthermore, all symmetric arcs of the complete bipartite graph with partitions $\{b, c\}$ and IS_n^1 are added. Similarly, all symmetric arcs of the complete bipartite graph with partitions

$\{d,e\}$ and IS_n^2 are added. Finally, the symmetric arcs $(a,b),(a,c),(d,f),(e,f)$ are added. The general shape of SD_n is depicted in Figure 5(a). The digraph SD_5 is represented in Figure 5(b).

Note that the set $F = \{y_1,\dots,y_n,b,c,d,e\}$ is a feedback vertex set of SD_n, with $|F| = n+4$. Thus $mfvs(SD_n) \leq n+4$ (actually, one can easily check that F is a minimum feedback vertex set of SD_n). Clearly, $pn(SD_n) \geq n$. In what follows, we prove that any strategy using $n+1$ agents needs to cover at least $3n+2$ vertices, and we present a $(n+1,3n+2)$-process strategy for SD_n. Since $mfvs_{n+1}(D) \leq mfvs_{pn}(D)$ for any digraph D, the result follows.

First, we prove by contradiction that all process strategies for SD_n using $n+1$ agents must start by processing either the nodes b and c or the nodes d and e, and so by placing the $n+1$ agents either at vertices a and $x_1,\dots,x_n$ or at vertices f and $z_1,\dots,z_n$.

Suppose that the first vertex to be processed is either a or belongs to IS_n^1, and it is processed at step s. Therefore, the vertices b and c must be occupied by agents at this step (such that a can be processed thereafter). Without loss of generality, let us assume that b is processed, say at step s', before c. Since at most $n-1$ agents are available while c and b are occupied, no vertex of the clique K_{n+1} can be processed before step s'. On the other hand, at step s', all vertices of IS_n^1 are processed or occupied by agents such that b can be processed. Let X be the subset of vertices of IS_n^1 that are occupied at step s', and let $Y = V(IS_n^1) \setminus X$. For any $x_i \in Y$, y_i must be occupied at step s' (since x_i is processed and y_i is not). Hence, at step s', at least $2+|X|+|Y| = n+2$ agents are occupying some vertices, a contradiction. By symmetry, f and any vertex of IS_n^2 cannot be the first vertex to be processed.

Now suppose that the first vertex to be processed is $y_i \in K_{n+1}$, $i \leq n+1$. Note that all vertices of K_{n+1}, but $y_{n+1} = v$, have at least $n+2$ outneighbors. Therefore, $i = n+1$. When v is processed, the n vertices of $K_{n+1} \setminus \{v\}$ must be occupied, leaving at most one free agent. But now, all vertices of K_{n+1} but v have at least 2 unprocessed outneighbors. Whatever be the placement of the last agent, no other vertex can be processed and no agents can be released. Hence, the strategy fails, a contradiction.

Hence, any process strategy using $n+1$ agents must start by processing b,c,d or e. Without loss of generality, (by symmetry), let us assume that the first vertex to be processed is b. Hence, the strategy must start by placing agents at any vertex in $\{a\} \cup V(IS_n^1)$. At this step, the strategy processes b and c without covering them. Then a can be processed and the agent at it is released. At this step, no other vertex can be processed. Moreover, the only move that can be done is to place the free agent at y_n. Indeed, any other move would let all agents blocked. Then the free agent is placed at node y_n and x_n can be processed and the agent occupying it can be released. Similarly, the strategy sequentially places an agent at y_{n-i}, processes

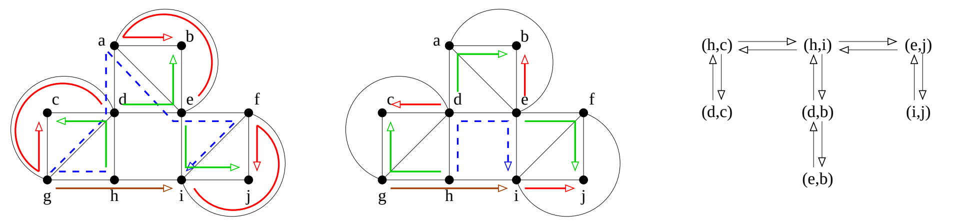
x_{n-i} and removes the corresponding agent, for $1 \leq i \leq n-1$. It is easy to check that any variation of this would make the strategy immediately fail. Once all vertices $y_1, \cdots, y_n$ are occupied, then v can be processed without being covered. Then, the strategy goes on being highly constrained: for $1 \leq i \leq n$, the free agent occupies z_i, allowing to process y_i and to free the agent occupying it. Finally, when all vertices of IS_n^2 are occupied, the free agent must occupy f, and all remaining vertices may be processed. Again, all these moves are forced for, otherwise, the strategy would be blocked.

Such a strategy covers $3n+2$ nodes. Therefore, $mfvs_{pn}(SD_n) \geq mfvs_{n+1}(SD_n) = 3n+2$. Hence, $\frac{mfvs_{pn}(SD_n)}{mfvs(SD_n)} \geq \frac{3n+2}{n+4}$. For $n > \frac{10}{\varepsilon} - 4$, we get $\frac{mfvs_{pn}(SD_n)}{mfvs(SD_n)} \geq 3 - \varepsilon$. Moreover, since SD_n has $3n+7$ vertices, we get $mfvs_{pn}(SD_n) \leq 3n+6$, and so $\frac{mfvs_{pn}(SD_n)}{mfvs(SD_n)} < 3$. □

Conjecture 1. For any symmetric digraph D, $\frac{mfvs_{pn}(D)}{mfvs(D)} \leq 3$.

5. Process Strategy out of the Routing Reconfiguration Problem

The *routing reconfiguration problem* occurs in connection-oriented networks such as telephone, MPLS, or WDM [3, 5–7, 19, 20]. In such networks, a connection corresponds to the transmission of a data flow from a source to a destination, and is usually associated with a capacited path (or a wavelength in WDM optical networks). A *routing* is the set of paths serving the connections. To avoid confusion, we assume here that each arc of the network has capacity one, and that each connection requires one unit of capacity. Consequently, no two paths can share the same arc (valid assumption in WDM networks). When a link of the network needs to be repaired, it might be necessary to change the routing of the connection using it, and incidentally to change the routing of other connections if the network has not enough free resources. Computing a new viable routing is a well known hard problem, but it is not the concern of this paper. Indeed, this is not the end of our worries: once a new routing not using the unavailable links is computed, it is not acceptable to stop all the connections going on, and change the routing, as it would result in a bad quality of service for the users (such operation requires minutes in WDM networks). Instead, it is preferred that each connection first establishes the new path on which it transmits data, and then stops the former one. This requires a proper scheduling to avoid conflicts in accessing resources (resources needed for a new path must be freed by other connections first). Furthermore, cyclic dependencies might force to interrupt some connections during that phase. The aim of the routing reconfiguration problem is to optimize tradeoffs between the total number and the concurrent number of connections to interrupt.



(a) Old set of routes S_1.

(b) New set of routes S_2.

(c) Dependency digraph D from S_1 to S_2.

Figure 6: Instance of the reconfiguration problem consisting of a network with 10 nodes and symmetric arcs, 8 connections $(h,i),(h,c),(d,c),(d,b),(e,b),(e,j),(i,j),(g,i)$ to be reestablished. Figure 6(a) depicts the old set of routes S_1, Figure 6(b) the new set S_2, and Figure 6(c) the dependency digraph from S_1 to S_2.

As an example, a way to reconfigure the instance depicted in Figure 6 may be to interrupt connections $(h,c),(d,b),(e,j)$, then set up the new paths of all other connections, tear down their old routes, and finally, set up the new paths of connections $(h,c),(d,b),(e,j)$. Such a strategy interrupts a total of 3 connections and these ones are interrupted simultaneously. Another strategy may consist of interrupting the connection (h,i), then sequentially: interrupt connection (h,c), reconfigure (d,c) without interruption for it, set up the new route of (h,c), then reconfigure in the same way first (d,b) and (e,b) without interruption for these two requests, and then (e,j) and (i,j). Finally, set up the new route of (h,i). The second strategy implies the interruption of 4 connections, but at most 2 connections are interrupted simultaneously.

Indeed, possible objectives are (1) to minimize the maximum number of concurrent interruptions [5, 6, 19, 20], and (2) to minimize the total number of disrupted connections [13]. Following [6, 13], these two problems can be expressed through the theoretical game described in Section 1.1, on the dependency digraph [13]. Given the initial routing and the new one, the dependency digraph contains one node per connection that must be switched. There is an arc from node u to node v if the initial route of connection v uses resources that are needed by the new route of connection u. Figure 6 shows an instance of the reconfiguration problem and its corresponding dependency digraph. In Figure 6(c), there is an arc from vertex (d,c) to vertex (h,c), because the new route used by connection (d,c) (Figure 6(b)) uses resources seized by connection (h,c) in the initial configuration (Figure 6(a)). Other arcs are built in the same way.

Given the dependency digraph D of an instance of the problem, a (p,q)-process strategy for D corresponds to a valid reconfiguration of the connections where p

is the maximum number of concurrent disruptions and q is the total number of interruptions. Indeed the three rules can be viewed in terms of reconfiguration of requests:

R_1 Put an agent at a vertex v of D;
Interrupt the request corresponding to v;

R_2 Remove an agent from a vertex v of D if all its outneighbors are either processed or occupied by an agent, and process v;
Route an interrupted connection when final resources are available;

R_3 Process an unoccupied vertex v of D if all its outneighbors are either processed or occupied by an agent;
Reroute a non-interrupted connection when final resources are available.

The next theorem proves the equivalence between instances of the reconfiguration problem and dependency digraphs.

Theorem 7. *Any digraph D is the dependency digraph of an instance of the routing reconfiguration problem whose network is a grid.*

Proof. Roughly, consider a grid network where each initial lightpath of any connection is some row of the grid. If two connections i and k are linked by an arc (i,k) in the dependency digraph, then we build the new lightpaths of both connections as depicted in Figure 7 which actually create the desired dependence. Note that the lightpath of connection k is deported on an additional row, i.e., a row corresponding to no connection. For each arc of the dependency digraph, we can use different columns of the grid-network, in such a way that these transformations may be done independently.

More formally, let $D = (V,A)$ be a digraph with $V = \{c_1, \cdots, c_n\}$ and $A = \{a_1, \cdots, a_m\}$. Let us define the network G as a $(n+2) \times (2m)$ grid such that each edge of which has capacity one. Let R_i denotes the i^{th} row of G ($0 \leq i \leq n+1$) and C_i its i^{th} column ($1 \leq j \leq 2m$), and let $v_{i,j} \in V(G)$ be the vertex in $R_i \cap C_j$. For any i, $1 \leq i \leq n$, connection i, corresponding to c_i in D, occurs between $v_{i,1} \in V(G)$ the leftmost vertex of R_i and $v_{i,2m} \in V(G)$ the rightmost vertex of R_i, and let the initial lightpath of connection i follows R_i. Now, we present an iterative method to build the new lightpath of each connection. Initially, for any i, $1 \leq i \leq n$, the new lightpath P_i^0 of connection i equals the old lightpath R_i. Now, after the $(j-1)^{th}$ step ($0 < j \leq m$) of the method, let P_i^{j-1} be the current value of the new lightpath of connection i and assume that in the subgraph of G induced by columns $(C_{2j-1}, \cdots, C_{2m})$, P_i^{j-1} equals R_i. Consider $a_j = (c_i, c_k) \in A$ and let us do

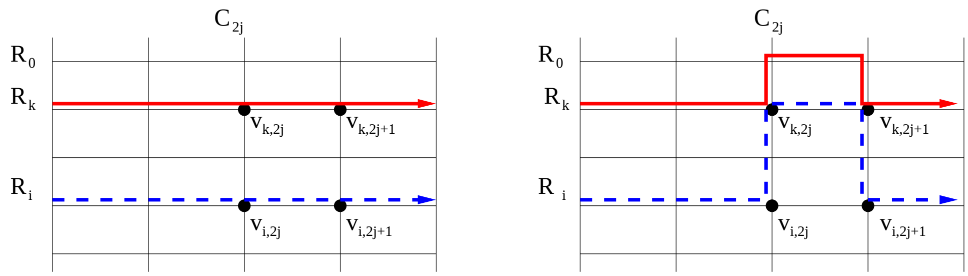


Figure 7: Scheme of the transformation in the proof of Theorem 7

the following transformation depicted in Figure 7. For any $\ell \notin \{i,k\}$, $P_\ell^j = P_\ell^{j-1}$. Now, P_i^j is defined by replacing the edge $(v_{i,2j-1}, v_{i,2j})$ in P_i^{j-1} by the shortest path from $v_{i,2j-1}$ to $v_{k,2j-1}$ (following C_{2j-1}), the edge $(v_{k,2j-1}, v_{k,2j})$, and the shortest path from $v_{k,2j}$ to $v_{i,2j}$ (following C_{2j}). Similarly, P_k^j is defined by replacing the edge $(v_{k,2j-1}, v_{k,2j})$ in P_k^{j-1} by the shortest path from $v_{k,2j-1}$ to $v_{n+1,2j-1}$ if $i < k$ (resp., to $v_{0,2j-1}$ if $i > k$), the edge $(v_{n+1,2j-1}, v_{n+1,2j})$ (resp., $(v_{0,2j-1}, v_{0,2j})$), and the shortest path from $v_{n+1,2j}$ to $v_{k,2j}$ (resp., from $v_{0,2j}$ to $v_{k,2j}$). It is easy to check that the grid G, the sets of initial lightpaths $\{R_1, \cdots, R_n\}$ and final lightpaths $\{P_1^m, \cdots, P_n^m\}$ admit D as dependency digraph. $\square$

Note that a digraph may be the dependency digraph of various instances of the reconfiguration problem. Since any digraph may be the dependency digraph of a realistic instance of the reconfiguration problem, Theorem 7 shows the relevance of studying these problems through dependency digraph notion.

A feasible reconfiguration may be defined by a (p,q)-process strategy for the corresponding dependency digraph. Problem (1) is equivalent to minimize p (Section 1.1) and Problem (2) is similar to the one of minimizing q (Section 1.1). Consider the dependency digraph D of Figure 6. From Section 1.1, we can not minimize both p and q, that is the number of simultaneous disrupted requests and the total number of interrupted connections. Indeed there does not exist a $(2,3)$-process strategy while $(2,4)$ and $(3,3)$ exist (Figure 1(a) and Figure 1(b)).

It is now easy to establish the relationship between tradeoff metrics introduced in Section 1.1 and tradeoffs for the routing reconfiguration problem. For example, pn_{mfvs} introduced in Definition 3 represents the minimum number of requests that have to be simultaneously interrupted during the reconfiguration when the total number of interrupted connections is minimum. Also Section 2 shows that the problems of computing these new tradeoffs parameters for the routing reconfiguration problem are NP-complete and not in APX. Finally Section 3 proves that the loss one can expect on one parameter when minimizing the other may be arbitrarily large.

6. Conclusion

In this paper, we address the routing reconfiguration problem through a game played on digraphs. We introduce the notion of (p,q)-process strategy and some tradeoff metrics in order to minimize one metric under the constraint that the other is fixed. We proved that the problems of computing these parameters are APX-hard and some are not in APX. We also proved that there exist digraphs for which minimizing one parameter may increase the other arbitrarily. For further research, we plan to continue our study for symmetric digraphs in order to (dis)prove Conjecture 1. Moreover, it would be interesting to design exact algorithms and heuristics to compute (p,q)-process strategies.

Acknowledgments

This work has been partially supported by région PACA, ANRs AGAPE and DIMAGREEN.

References

[1] Bafna, V., Berman, P., Fujito, T., 1999. A 2-approximation algorithm for the undirected feedback vertex set problem. SIAM J. Discrete Math 12, 289297.

[2] Breisch, R. L., 1967. An intuitive approach to speleotopology. Southwestern Cavers VI (5), 72–78.

[3] Cohen, N., Coudert, D., Mazauric, D., Nepomuceno, N., Nisse, N., Jun. 2010. Tradeoffs in process strategy games with application in the WDM reconfiguration problem. In: Boldi, P., Gargano, L. (Eds.), Fifth International conference on Fun with Algorithms (FUN 2010). Vol. 6099 of Lecture Notes in Computer Science. Springer, Ischia Island, Italy, pp. 121–132.

[4] Coudert, D., Huc, F., Mazauric, D., Sep. 2008. A distributed algorithm for computing and updating the process number of a forest. In: Taubenfeld, G. (Ed.), 22nd International Symposium on Distributed Computing (DISC). Vol. 5218 of Lecture Notes in Computer Science. Springer, Arcachon, France, pp. 500–501, brief announcement.

[5] Coudert, D., Huc, F., Mazauric, D., Nisse, N., Sereni, J.-S., 2009. Routing reconfiguration/process number: Coping with two classes of services. In: 13th Conf. on Optical Network Design and Modeling (ONDM).

[6] Coudert, D., Perennes, S., Pham, Q.-C., Sereni, J.-S., mai 2005. Rerouting requests in WDM networks. In: 7ème Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel'05). pp. 17–20.

[7] Coudert, D., Sereni, J.-S., Sep. 2007. Characterization of graphs and digraphs with small process number. Research Report 6285, INRIA.

[8] Deo, N., Krishnamoorthy, S., Langston, M. A., 1987. Exact and approximate solutions for the gate matrix layout problem. IEEE Tr. on Comp.-Aided Design 6, 79–84.

[9] Díaz, J., Petit, J., Serna, M., 2002. A survey on graph layout problems. ACM Comp. Surveys 34 (3), 313–356.

[10] Ellis, J., Sudborough, I., Turner, J., 1994. The vertex separation and search number of a graph. Information and Computation 113 (1), 50–79.

[11] Even, G., Naor, J., Schieber, B., , Sudan, M., 1995. Approximating minimum feedback sets and multi-cuts in directed graphs. In: Proc. 4th Int. Conf. on Integer Prog. and Combinatorial Optimization. Vol. LNCS 920.

[12] Fomin, F., Thilikos, D., 2008. An annotated bibliography on guaranteed graph searching. Theo. Comp. Sci. 399 (3), 236–245.

[13] Jose, N., Somani, A., Oct. 2003. Connection rerouting/network reconfiguration. In: 4th International Workshop on Design of Reliable Communication Networks (DRCN). IEEE, pp. 23–30.

[14] Kann, V., 1992. On the approximability of NP-complete optimization problems. Ph.D. thesis, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm., see also: http://www.csc.kth.se/ viggo/wwwcompendium/node19.html.

[15] Kirousis, M., Papadimitriou, C., 1986. Searching and pebbling. Theoretical Comp. Sc. 47 (2), 205–218.

[16] Megiddo, N., Hakimi, S. L., Garey, M. R., Johnson, D. S., Papadimitriou, C. H., 1988. The complexity of searching a graph. J. Assoc. Comput. Mach. 35 (1), 18–44.

[17] Parsons, T. D., 1978. Pursuit-evasion in a graph. In: Theory and applications of graphs. Vol. 642 of Lecture Notes in Mathematics. Springer, Berlin, pp. 426–441.

[18] Robertson, N., Seymour, P. D., 1983. Graph minors. I. Excluding a forest. J. Comb. Th. Ser. B 35 (1), 39–61.

[19] Solano, F., 2009. Analyzing two different objectives of the WDM network reconfiguration problem. In: IEEE Global Communications Conference (Globecom).

[20] Solano, F., Pióro, M., 2009. A mixed-integer programing formulation for the lightpath reconfiguration problem. In: VIII Workshop on G/MPLS Networks (WGN8).

Good edge-labelling of graphs☆

J. Araujo[a,b], N. Cohen[a], F. Giroire[a], F. Havet[a]

[a] *Projet Mascotte I3S (CNRS & Univ. de Nice) and INRIA, INRIA Sophia Antipolis, 2004 route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex France.*
[b] *ParGO Research Group, Federal University of Ceará, Campus do Pici, Bloco 910, 60455-760 - Fortaleza - Brazil.*

Abstract

A *good edge-labelling* of a graph G is a labelling of its edges such that, for any ordered pair of vertices (x, y), there do not exist two paths from x to y with increasing labels. This notion was introduced in [1] to solve wavelength assignment problems for specific categories of graphs. In this paper, we aim at characterizing the class of graphs that admit a good edge-labelling. First, we exhibit infinite families of graphs for which no such edge-labelling can be found. We then show that deciding if a graph G admits a good edge-labelling is NP-complete, even if G is bipartite. Finally, we give large classes of graphs admitting a good edge-labelling: C_3-free outerplanar graphs, planar graphs of girth at least 6, subcubic graphs.

Keywords: Graph Theory, NP-completeness, Edge-labelling, Increasing paths.

1. Introduction

A classical and widely studied problem in WDM (Wavelength Division Multiplexing) networks is the Routing and Wavelength Assignment (RWA) problem [2, 3, 4]. It consists in finding routes, and their associated wavelength as well, to satisfy a set of traffic requests while minimizing the number of used wavelengths. This is a difficult problem which is, in general, NP-hard. Thus, it is often split into two distinct problems: First, routes are found for the requests. Then, in a second step, these routes are taken as an input. Wavelengths must be associated to them in such a way that two routes using the same fiber do not have the same wavelength. The last problem can be reformulated as follows: Given a digraph and a set of dipaths, corresponding to the routes for the requests, find the minimal number of wavelengths w needed to assign different

☆Partially supported by the INRIA associated team EWIN between Mascotte and the team ParGO of the Federal University of Ceará, and by the ANR Blanc International Taiwan GRATEL.

Email addresses: `julio.araujo@sophia.inria.fr` (J. Araujo), `nathann.cohen@sophia.inria.fr` (N. Cohen), `frederic.giroire@sophia.inria.fr` (F. Giroire), `frederic.havet@sophia.inria.fr` (F. Havet)

wavelengths to dipaths sharing an edge. This problem can be seen as a colouring problem of the *conflict graph* which is defined as follows: It has one vertex per dipath and two vertices are linked by an edge if their corresponding dipaths share an edge. In [1], Bermond et al. studied the RWA problem for UPP-DAG which are acyclic digraphs (or DAG) in which there is at most one dipath from one vertex to another. In such digraph the routing is forced and thus the unique problem is the wavelength assignment one.

In their paper, they introduce the notion of good edge-labelling. An *edge-labelling* of a graph G is a function $\phi : E(G) \to \mathbb{R}$. A path is *increasing* if the sequence of its edge labels is non-decreasing. An edge-labelling of G is *good* if, for any two distinct vertices u, v, there is at most one increasing (u, v)-path. Bermond et al. [1] showed that the conflict graph of a set of dipaths in a UPP-DAG has a *good edge-labelling*. Conversely, for any graph admitting a good edge-labelling one can exhibit a family of dipaths on a UPP-DAG whose conflict graph is precisely this graph. Bermond et al. [1] then use the existence of graphs with a good edge-labelling and large chromatic number to prove that there exist sets of requests on UPP-DAGs with load 2 (an edge is shared by at most two paths) requiring an arbitrarily large number of wavelengths.

To obtain other results on this problem, it may be useful to identify the *good* graphs which admit a good edge-labelling and the *bad* ones which do not. Bermond et al. [1] noticed that C_3 and $K_{2,3}$ are bad. J.-S. Sereni [5] asked whether every $\{C_3, K_{2,3}\}$*-free graph* (i.e., with no C_3 nor $K_{2,3}$ as a subgraph) is good. In Section 3, we answer this question in the negative. We give an infinite family of bad graphs none of which is the subgraph of another.

Furthermore, in Section 4, we prove that determining if a graph has a good edge-labelling is NP-complete using a reduction from Not-All-Equal 3-SAT.

In Section 5, we show large classes of good graphs: forests, C_3-free outerplanar graphs, planar graphs of girth at least 6. To do so, we use the notion of *critical* graph which is a bad graph such that every proper subgraph of which is good. Clearly, a good edge-labelling of a graph induces a good edge-labelling of all its subgraphs. So every bad graph must contain a critical subgraph. We establish several properties of critical graphs. In particular, we show that they have no *matching-cut*. Hence, a result of Farley and Proskurowski [6] (Theorem 16) implies that a critical graph G has at least $\frac{3}{2}|V(G)| - \frac{3}{2}$ edges.

In Section 6, we use the characterization of graphs with no matching-cut and $\left\lceil \frac{3}{2}|V(G)| - \frac{3}{2} \right\rceil$ edges given by Bonsma [7] to slightly improve this result. We show that a critical graph G has at least $\frac{3}{2}|V(G)| - \frac{1}{2}$ edges unless G is C_3 or $K_{2,3}$.

Finally, we present avenues for future research.

2. Preliminaries

In this section, we give some technically useful propositions. Their proofs are straightforward and left to the reader.

A path is *decreasing* if the sequence of its edge labels is non-increasing. Then, a path $u_1u_2 \dots u_k$ is decreasing if and only if its reversal $u_ku_{k-1} \dots u_1$ is increasing. Hence an edge-labelling is good if and only if for any two distinct vertices u, v, there is at most one decreasing (u, v)-path. Equivalently, an edge-labelling is good if and only if for any two distinct vertices u, v, there is at most one increasing (u, v)-path and at most one decreasing (u, v)-path.

Let x and y be two vertices of G. Two distinct (x, y)-paths P and Q are *independent* if $V(P) \cap V(Q) = \{x, y\}$. Observe that in an edge-labelled graph G, there are two vertices u, v with two increasing (u, v)-paths if and only if there are two vertices u', v' with two increasing independent (u', v')-paths. Hence the definition of good edge-labelling may be expressed in terms of independent paths.

Proposition 1. *An edge-labelling is good if and only if for any two distinct vertices u and v, there are no two increasing independent (u, v)-paths.*

As above the definition may also be in terms of decreasing independent paths. In the paper, we sometimes use Proposition 1 without referring explicitly to it.

Let ϕ be a good edge-labelling of a graph G. If $\phi(E(G)) \subset A$ then for every strictly increasing function $f : A \to B$, $f \circ \phi$ is a good edge-labelling into B. Moreover if ϕ is not injective, one can transform it into an injective one by recursively adding a tiny ϵ to one of the edges having the same label. Hence we have the following.

Proposition 2. *Let G be a graph and A an infinite set in $\mathbb{R} \cup \{-\infty, +\infty\}$. Then G admits a good edge-labelling if and only if it admits an injective good edge-labelling into A.*

3. Bad graphs

A path of length one is both increasing and decreasing, and a path of length two is either increasing or decreasing. So C_3 has clearly no good edge-labelling. Also $K_{2,3}$ does not admit a good edge-labelling since there are three paths of length two between the two vertices of degree 3. Hence, in any edge-labelling, two of them are increasing or two of them are decreasing.

Extending this idea, we now construct an infinite family of bad graphs, none of which is the subgraph of another. The construction of this family is based on the graphs H_k defined below. These graphs play the same role as a path of length two because they have two vertices u and v such that any good edge-labelling of H_k has either a (u, v)-increasing path or a (v, u)-increasing path.

For any integer $k \geq 3$, let H_k be the graph defined by

$$\begin{aligned} V(H_k) &= \{u, v\} \cup \{u_i \mid 1 \leq i \leq k\} \cup \{v_i \mid 1 \leq i \leq k\}, \\ E(H_k) &= \{uu_i \mid 1 \leq i \leq k\} \cup \{u_i v_i \mid 1 \leq i \leq k\} \cup \{v_i v \mid 1 \leq i \leq k\}, \\ &\quad \cup \{v_i u_{i+1} \mid 1 \leq i \leq k\} \end{aligned}$$

with $u_{k+1} = u_1$. See Figure 1.

Observe that the graph H_k has no $K_{2,3}$ as a subgraph, and for $i \neq k$, H_i is not a subgraph of H_k.

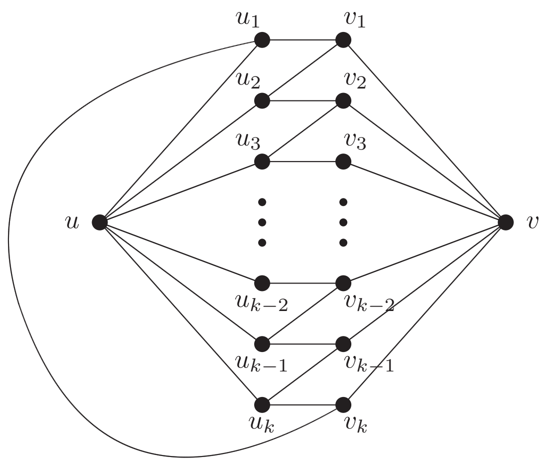


Figure 1: Graph H_k

Proposition 3. *Let $k \geq 3$. For every good edge-labelling, the graph H_k has either an increasing (u, v)-path or an increasing (v, u)-path.*

Proof: Suppose, by way of contradiction, that H_k has a good edge-labelling ϕ having no increasing (u, v)-path and no increasing (v, u)-path. By Proposition 2, we may assume that ϕ is injective.

A key component in this proof is the following observation which follows easily from the fact that ϕ is good.

Observation 3.1. *Suppose $x_1x_2x_3x_4x_1$ is a 4-cycle. Then, either*

- $\phi(x_4x_1) < \phi(x_1x_2)$, $\phi(x_2x_3) < \phi(x_1x_2)$, $\phi(x_2x_3) < \phi(x_3x_4)$ *and* $\phi(x_1x_4) < \phi(x_3x_4)$*; or*
- *all those inequalities are reversed.*

By symmetry, we may assume that $\phi(uu_1) < \phi(u_1v_1)$. By Observation 3.1, $\phi(v_1u_2) < \phi(u_1v_1)$, $\phi(v_1u_2) < \phi(uu_2)$ and $\phi(uu_1) < \phi(uu_2)$. Then, since vv_1u_2u is not increasing, $\phi(u_2v_1) < \phi(v_1v)$. Again by Observation 3.1, $\phi(v_2v) < \phi(u_2v_2)$. Thus since uu_2v_2v is not increasing $\phi(uu_2) < \phi(u_2v_2)$.

Applying the same reasoning, we obtain that $\phi(uu_2) < \phi(uu_3)$ and $\phi(uu_3) < \phi(u_3v_3)$ and so on, iteratively, $\phi(uu_1) < \phi(uu_2) < \cdots < \phi(uu_k) < \phi(uu_1)$, a contradiction. □

For convenience we denote by H_2 the path of length 2 with end vertices u and v. Let i, j, k be three integers greater than 1. The graph $J_{i,j,k}$ is the graph obtained from disjoint copies of H_i, H_j and H_k by identifying the vertices u of the three copies and the vertices v of the three copies.

Proposition 4. *Let i, j, k be three integers greater than 1. Then $J_{i,j,k}$ is bad.*

Proof: Suppose, by way of contradiction, that $J_{i,j,k}$ admits a good edge-labelling. By Proposition 3, in each of the subgraphs H_i, H_j and H_k, there is either an increasing (u, v)-path or an increasing (v, u)-path. Hence in $J_{i,j,k}$, there are either two increasing (u, v)-paths or two increasing (v, u)-paths, a contradiction. □

4. *NP*-completeness

In this section, we prove that it is an NP-complete problem to decide if a bipartite graph admits a good edge-labelling. We give a reduction from the NOT-ALL-EQUAL (NAE) 3-SAT Problem [8] which is defined as follows:

Instance: A set V of variables and a collection $\mathcal{C}$ of clauses over V such that each clause has exactly 3 literals.

Question: Is there a truth assignment such that each clause has at least one true and at least one false literal?

For sake of clarity, we first present the NP-completeness proof for general graphs.

Theorem 5. *The following problem is NP-complete.*
Instance: *A graph G.*
Question: *Does G have a good edge-labelling?*

Proof: Given a graph G and an injective edge-labelling ϕ into $\mathbb{R}$, one can check in polynomial time if ϕ is good or not using the following algorithm where $(u_1v_1, \ldots, u_mv_m)$ is an ordering of the edges of G in increasing order according to their labels.

```
foreach u ∈ V(G) do
    Set V(T) := {u}, E(T) := ∅;
    foreach i=1 to m do
        if {u_i, v_i} ⊂ V(T) then
            return "bad edge-labelling";
        if u_i ∈ V(T) (and v_i ∉ V(T)) then
            V(T) := V(T) ∪ {v_i} and E(T) := E(T) ∪ {u_i v_i};
return "good edge-labelling";
```

Indeed, for each vertex u, the above algorithm grows the tree T of increasing paths from u: at each step i, T is the tree of increasing paths from u with arcs with labels less than $\phi(u_iv_i)$. In particular, there is an increasing (u, v)-path P_v for every $v \in V(T)$. Hence if $u_i \in V(T)$ and $v_i \in V(T)$ then P_{v_i} and $P_{u_i} + u_iv_i$ are two increasing (u, v_i)-paths, so the edge-labelling is not good. If $u_i \in V(T)$ and $v_i \notin V(T)$, then $P_{u_i} + u_iv_i$ is a new increasing path that must be included into T. Finally, if $u_i \notin V(T)$ and $v_i \notin V(T)$, then u_iv_i will not be in any increasing path from u as the edges to be considered after it have larger labels.

Hence the considered problem is in NP.

To prove that the problem is NP-complete, we will reduce the NAE 3-SAT Problem without repetition (i.e. a variable appears at most once in each clause) which is equivalent to NAE 3-SAT Problem (with repetition) to it. (For each repeated variable x, we introduce two other variables y and z. Then the second (third) occurrence of x in a clause is replaced by y (z). Then, x, y, z are forced to have the same truth assignment by adding $\bar{x} \vee y \vee z$, $x \vee \bar{y} \vee z$, $x \vee y \vee \bar{z}$, $\bar{x} \vee \bar{y} \vee z$, $\bar{x} \vee y \vee \bar{z}$, and $x \vee \bar{y} \vee \bar{z}$ to the instance.)

Let $V = \{x_1, \ldots, x_n\}$ and $\mathcal{C} = \{C_1, \ldots, C_m\}$ be an instance I of the NAE 3-SAT Problem without repetition. We shall construct a graph G_I in such a way that I has an answer yes for the NAE 3-SAT Problem if and only if G_I has a good edge-labelling.

For each variable x_i, $1 \le i \le n$, we create a variable graph VG_i defined as follows (See Figure 2.):

$$\begin{aligned}
V(VG_i) &= \{v_k^{i,j} \mid 1 \le j \le m, 1 \le k \le 4\} \cup \{r_k^{i,j} \mid 1 \le j \le m, 1 \le k \le 4\} \\
&\quad \cup \{s_k^{i,j} \mid 1 \le j \le m, 1 \le k \le 4\}. \\
E(VG_i) &= \{v_k^{i,j} v_{k+1}^{i,j} \mid 1 \le j \le m, 1 \le k \le 3\} \cup \{v_4^{i,j} v_1^{i,j+1} \mid 1 \le j \le m-1\} \\
&\quad \cup \{v_k^{i,j} r_k^{i,j} \mid 1 \le j \le m, 1 \le k \le 4\} \cup \{v_k^{i,j} s_k^{i,j} \mid 1 \le j \le m, 1 \le k \le 4\} \\
&\quad \cup \{v_4^{i,j} r_1^{i,j} \mid 1 \le j \le m\} \cup \{v_k^{i,j+1} r_{k+1}^{i,j} \mid 1 \le j \le m-1, 1 \le k \le 3\} \\
&\quad \cup \{v_4^{i,j} s_1^{i,j} \mid 1 \le j \le m\} \cup \{v_k^{i,j+1} s_{k+1}^{i,j} \mid 1 \le j \le m-1, 1 \le k \le 3\}.
\end{aligned}$$

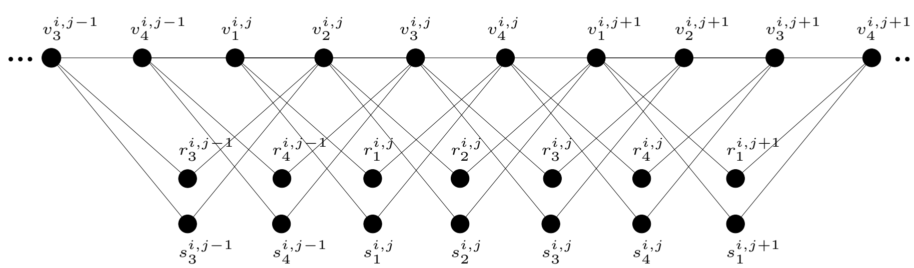


Figure 2: The variable graph VG_i

For each clause $C_j = l_1 \vee l_2 \vee l_3$, $1 \le j \le m$, we create a clause graph CG_j defined as follows (See Figure 3.):

$$\begin{aligned}
V(CG_j) &= \{c^j, b_1^j, b_2^j, b_3^j\}; \\
E(CG_j) &= \{c^j b_1^j, c^j b_2^j, c^j b_3^j\}.
\end{aligned}$$

Now, for each literal l_k, $1 \le k \le 3$, if l_k is the non-negated variable x_i, we identify b_k^j, c^j and b_{k+1}^j (index k is taken modulo 3) with $v_1^{i,j}$, $v_2^{i,j}$ and $v_3^{i,j}$,

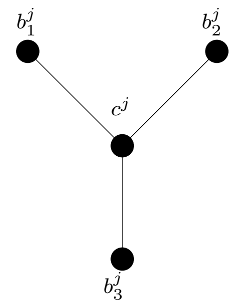


Figure 3: The clause graph CG_j.

respectively. Otherwise, if l_k is the negated variable $\bar{x}_i$, we identify b_k^j, c^j and b_{k+1}^j with $v_3^{i,j}$, $v_2^{i,j}$ and $v_1^{i,j}$, respectively.

. Let us first show that, if G_I has a good edge-labelling ϕ, then there is a truth assignment such that each clause of I has at least one true literal and at least one false literal.

By Proposition 2, we may assume that ϕ is injective.

Claim 5.1. *Let $1 \leq i \leq n$. If $\phi(v_1^{i,1}v_2^{i,1}) < \phi(v_2^{i,1}v_3^{i,1})$ then $\phi(v_1^{i,j}v_2^{i,j}) < \phi(v_2^{i,j}v_3^{i,j})$ for all $1 \leq j \leq m$.*

Proof: By induction on j. A path of length two is necessarily increasing or decreasing. Now $v_1^{i,j}$ is joined to $v_4^{i,j}$ by two paths of length two via $r_1^{i,j}$ and $s_1^{i,j}$. Since ϕ is good, one of these two paths is increasing and the other one is decreasing. In addition, the path $v_1^{i,j}v_2^{i,j}v_3^{i,j}v_4^{i,j}$ is neither increasing nor decreasing so $\phi(v_2^{i,j}v_3^{i,j}) > \phi(v_3^{i,j}v_4^{i,j})$.

Applying three times this reasoning, we derive $\phi(v_3^{i,j}v_4^{i,j}) < \phi(v_4^{i,j}v_1^{i,j+1})$, $\phi(v_4^{i,j}v_1^{i,j+1}) > \phi(v_1^{i,j+1}v_2^{i,j+1})$ and finally $\phi(v_1^{i,j+1}v_2^{i,j+1}) < \phi(v_2^{i,j+1}v_3^{i,j+1})$. $\square$

Hence we define the truth assignment Λ by $\Lambda(x_i) = true$ if $\phi(v_1^{i,1}v_2^{i,1}) < \phi(v_2^{i,1}v_3^{i,1})$ and $\Lambda(x_i) = false$ otherwise.

Let us show that each clause C_j has at least one true literal or one false literal. Set $C_j = l_1 \vee l_2 \vee l_3$. First observe that, by construction, for all $1 \leq k \leq 3$, l_k is true if $\phi(b_k^j c^j) < \phi(b_{k+1}^j c^j)$ and l_k is false otherwise. Now the three literals are not all true otherwise, $\phi(b_1^j c^j) < \phi(b_2^j c^j) < \phi(b_3^j c^j) < \phi(b_1^j c^j)$, a contradiction. And they are not all false, otherwise $\phi(b_1^j c^j) > \phi(b_2^j c^j) > \phi(b_3^j c^j) > \phi(b_1^j c^j)$, a contradiction. Hence C_j has at least one true literal and one false literal.

. Conversely, let us now show that if there is a truth assignment Λ such that each clause of I has at least one true literal and at least one false literal, then G_I has a good edge-labelling.

The idea is to find a good edge-labelling ϕ satisfying the following property ($\star$): If $\Lambda(x_i) = true$, $\phi(v_1^{i,j}v_2^{i,j}) < \phi(v_2^{i,j}v_3^{i,j})$ for all $1 \leq j \leq m$ and if $\Lambda(x_i) = false$, $\phi(v_1^{i,j}v_2^{i,j}) > \phi(v_2^{i,j}v_3^{i,j})$ for all $1 \leq j \leq m$.

Let $C_j = l_1 \vee l_2 \vee l_3$ be clause. To satisfy ($\star$), we must label the edges of VG_j such that $\phi(b_k^j c^j) < \phi(b_{k+1}^j c^j)$ if l_k is true and $\phi(b_k^j c^j) > \phi(b_{k+1}^j c^j)$ if l_k is false. As C_j has at least one true and one false literal, there is a unique way to label the three edges $c^j b_k^j$, $1 \leq k \leq 3$, with $\{-1, 0, +1\}$ such that the condition ($\star$) is fulfilled.

Let us now extend this edge-labelling to the remaining edges of each of the variable graphs VG_i. First, for all $1 \leq j \leq m$ and $1 \leq k \leq 4$, assign -3 and $+3$ alternatingly on the edges of the cycle of length four containing both $r_k^{i,j}$ and $s_k^{i,j}$ such that $\phi(v_k^{i,j}r_k^{i,j}) = -3$. Then, if $\Lambda(x_i) = true$, set $\phi(v_3^{i,j}, v_4^{i,j}) = -2$ and $\phi(v_4^{i,j}, v_1^{i,j+1}) = 2$ for all $1 \leq j \leq m$, and, if $\Lambda(x_i) = false$, set $\phi(v_3^{i,j}, v_4^{i,j}) = 2$ and $\phi(v_4^{i,j}, v_1^{i,j+1}) = -2$ for all $1 \leq j \leq m$.

We claim that ϕ is a good edge-labelling of G_I. Suppose, by way of contradiction, that there is a pair of vertices (x, y) such that two independent increasing (x, y)-paths P_1 and P_2 exist.

A set of two independent paths starting at a vertex of $R = \{r_k^{i,j} \mid 1 \leq j \leq m, 1 \leq k \leq 4\} \cup \{s_k^{i,j} \mid 1 \leq j \leq m, 1 \leq k \leq 4\}$ contains one increasing path (the one starting with the edge labelled -3) and one decreasing path (the one starting with the edge labelled 3). Hence x and y are not in R.

In addition, the union of P_1 and P_2 cannot be one of the four-cycles formed by the edges incident to $r_k^{i,j}$ and $s_k^{i,j}$ for some i,j and k.

Without loss of generality, we may assume that P_1 is at least as long as P_2. As cycles formed by two graphs GV_i and GV_j are of length at least 6, P_1 has length at least 3. Now one can see that P_1 may not contain any vertex of R because every path of length at least 3 with internal vertices in R is not increasing (nor decreasing).

Hence P_1 must contain at least three consecutive edges on one of the paths $Q_i = VG_i - R$. So P_1 is not increasing, a contradiction. $\square$

Observe that the graph G_I constructed in the above proof is not bipartite. However, with a slight modification, we can transform it into a bipartite graph and obtain the following theorem.

Theorem 6. *The following problem is NP-complete.*
Instance: *A bipartite graph G.*
Question: *Does G have a good edge-labelling?*

Proof: Let G'_I be the graph obtained from G_I (described in the proof of Theorem 5) by replacing each path $v_k^{i,j}, r_k^{i,j}, v_{k+3}^{i,j}$ and each path $v_k^{i,j}, s_k^{i,j}, v_{k+3}^{i,j}$, by copies of a graph $H_{k'}$ defined in Section 4, for some $k' \geq 3$ and for all $i = 1, \ldots, n$, $j = 1, \ldots, m$ and $k = 1, \ldots, 4$ ($k + 3$ is taken modulo 4).

By Proposition 3, it is not difficult to verify that G'_I admits a good edge-labelling if, and only if, G'_I also does. Moreover, each $H_{k'}$ admits a proper

2-colouring such that the vertices u and v have disjoint colours. Thus, G'_I is bipartite, since it admits a proper 2-colouring where all the vertices $v_1^{i,j}$ and $v_3^{i,j}$ belong to the same colour class, for all $i = 1, \ldots, n$ and $j = 1, \ldots, m$. □

5. Classes of good graphs

Recall that a graph G is *critical* if it is bad but each of its proper subgraphs is good. To prove that every graph in a class $\mathcal{C}$ closed under taking subgraphs has a good edge-labelling, it suffices to prove that $\mathcal{C}$ contains no critical graph.

Lemma 7. *Let G be a graph with a cutvertex x, $C_1, \ldots, C_k$ be the components of $G - x$ and $G_i = G\langle C_i \cup \{x\}\rangle$, $1 \leq i \leq k$. Then G is good if and only if all the G_i are good.*

Proof: Necessity is obvious since a good edge-labelling of G induces a good edge-labelling on each subgraph G_i.

Sufficiency follows from the fact that there are two independent (u, v)-paths in G only if there exists i, $1 \leq i \leq k$, such that u and v are in $V(G_i)$. Hence the union of good edge-labellings of all the G_i is a good edge-labelling of G. □

Corollary 8. *Every critical graph is* 2*-connected.*

Corollary 9. *Every forest F admits a good edge-labelling.*

Proof: No forest contains a non-trivial 2-connected subgraph, and so contains no critical subgraph. □

Let $G = (V, E)$ be a graph. A *K_2-cut* of G is a set of two adjacent vertices u and v such that the graph $G - \{u, v\}$ (obtained from G by removing u and v and their incident edges) has more connected components than G.

Lemma 10. *Let G be a connected graph and $\{u, v\}$ a K_2-cut in G such that $G-\{u, v\}$ has two connected components C_1 and C_2. If $G_1 = G\langle C_1 \cup \{u, v\}\rangle$ and $G_2 = G\langle C_2 \cup \{u, v\}\rangle$ have a good edge-labelling then G has a good edge-labelling.*

Proof: Let ϕ_1 and ϕ_2 be good edge-labellings of $G\langle C_1 \cup \{u, v\}\rangle$ and $G\langle C_2 \cup \{u, v\}\rangle$ respectively such that $\phi_1(uv) = \phi_2(uv)$.

Then the union of ϕ_1 and ϕ_2 is a good edge-labelling of G. Indeed, suppose by way of contradiction, that there exists x and y and two independent increasing (x, y)-paths P_1 and P_2 in G. W. l. o. g., we may assume that $x \in V(G_1)$. At least one of the paths, say P_1, contains at least one edge e_1 in $E(G_2) \setminus \{uv\}$ since ϕ_1 is good.

Assume first that $y \in V(G_1)$. Then P_1 must go through u and v. Let Q_2 be the shortest (u, v)-subpath of P_1 containing e_1. Then Q_2 is either increasing or decreasing. Hence since uv is both increasing and decreasing, there are either two increasing paths or two decreasing paths in G_2. This contradicts the fact that ϕ_2 is good.

Assume now that $y \in C_2$. Then since P_1 and P_2 are independent without loss of generality, P_1 goes through u and P_2 goes through v. Let Q_1 be the (x,u)-subpath of P_1, R_1 be the (u,y)-subpath of P_1, let Q_2 be the (x,v)-subpath of P_2 and R_2 be the (v,y)-subpath of P_2.

If $\phi(uv)$ is larger than the label of the last edge of Q_1, then Q_1uv and Q_2 are two increasing (x,v)-paths in G_1, a contradiction. If not $\phi(uv)$ is smaller than the label of the first edge of R_1 and vuR_1 and R_2 are two increasing (v,y)-paths in G_2, a contradiction.

$\square$

Let $G=(V,E)$ be a graph. An *edge-cut* is a non-empty set of edges between a set of vertices S and its complement $\overline{S}$. Formally, for any $S \subset V$, the edge-cut $[S,\overline{S}]$ is the set $\{uv \in E \mid u \in S \text{ and } v \in \overline{S})$. An edge cut which is also a matching is called a *matching-cut*.

Lemma 11. *Let G be a graph and $[S,\overline{S}]$ a matching-cut in G. If $G\langle S\rangle$ and $G\langle\overline{S}\rangle$ have a good edge-labelling then G has a good edge-labelling.*

Proof: Let ϕ_1 be a good edge-labelling of $G\langle S\rangle$ and ϕ_2 be a good edge-labelling of $G\langle\overline{S}\rangle$ (in $\mathbb{R}$). Then the edge-labelling ϕ of G defined by $\phi(e)=\phi_1(e)$ if $e \in E(G\langle S\rangle)$, $\phi(e)=\phi_2(e)$ if $e \in E(G\langle\overline{S}\rangle)$ and $\phi(e)=+\infty$ if $e \in [S,\overline{S}]$ is good.

Indeed, suppose by way of contradiction, that it is not good. Then there exist two vertices u and v and two independent increasing (u,v)-paths P_1 and P_2. Since ϕ_1 and ϕ_2 are good, then without loss of generality, we may assume that $u \in S$ and $v \in \overline{S}$. For $i=1,2$ P_i contains an edge of v_iw_i in $[S,\overline{S}]$. Now as v_1w_1 and v_2w_2 are labelled $+\infty$ and incident to no edges labelled $+\infty$, v_1w_1 must be the last edge of P_1 and v_2w_2 the last edge of P_2. So $w_1=v=w_2$ which is impossible as $[S,\overline{S}]$ is a matching. $\square$

Corollary 12. *A critical graph has no matching-cut.*

Corollary 13. *Every C_3-free outerplanar graph admits a good edge-labelling.*

Proof: An easy result of Eaton and Hull [9] states that a C_3-free outerplanar graph has either a vertex of degree 1 or two adjacent vertices of degree 2. This implies that it has a matching-cut. Hence by Corollary 12 no C_3-free outerplanar graph is critical, which yields the result. $\square$

A graph is *subcubic* if every vertex has degree at most three.

Lemma 14. *Every subcubic $\{C_3, K_{2,3}\}$-free graph has a matching-cut.*

Proof: Let G be a subcubic graph $\{C_3, K_{2,3}\}$-free. If G has no cycle, then every edge forms a matching-cut. Suppose now that G has a cycle. Let C be a cycle of smallest length in G. If C is a connected component of G (in particular if $C = G$) then any pair of non-adjacent edges of C forms a matching-cut.

If not, let us show that $[V(C), \overline{V(C)}]$ is a matching-cut. Let $e_1 = x_1y_1$ and $e_2 = x_2y_2$ be two distinct edges in this set with $x_1, x_2 \in V(C)$. Then $x_1 \neq x_2$ because these two vertices have degree (at most) 3 and they have two neighbours in $V(C)$. Suppose by way of contradiction that $y_1 = y_2$. Then x_1 and x_2 are not adjacent since G is C_3-free. Furthermore, there are the two (x_1, x_2)-paths along C are of length at most 2 otherwise C would not be a smallest cycle. Hence C is a cycle of length 4 and the graph induced by $V(C) \cup \{y_1\}$ is a $K_{2,3}$, a contradiction. □

Corollary 12 and Lemma 14 immediately imply that the sole subcubic critical graphs are C_3 and $K_{2,3}$.

Corollary 15. *Every subcubic $\{C_3, K_{2,3}\}$-free graph has a good edge-labelling.*

Farley and Proskurowski [6] proved that every (multi)graph G on n vertices with less than $\frac{3}{2}(n-1)$ edges has a matching-cut.

Theorem 16 (Farley and Proskurowski [6]). *Let G be a multigraph. If $|E(G)| < \frac{3}{2}|V(G)| - \frac{3}{2}$ then G has a matching-cut.*

Corollary 12 and Theorem 16 yield immediately the following.

Corollary 17. *Every critical graph has at least $\left\lceil \frac{3}{2}|V(G)| - \frac{3}{2} \right\rceil$ edges.*

An easy and well-known consequence of Euler's Formula states that every planar graph with girth at least 6 has at most $\frac{3}{2}|V(G)| - 3$ edges and so is not critical.

Corollary 18. *Every planar graph of girth at least* 6 *has a good edge-labelling.*

6. Good edge-labelling of ABC-graphs

Corollary 17 states that every critical graph has at least $\left\lceil \frac{3}{2}|V(G)| - \frac{3}{2} \right\rceil$ edges. This is tight since if G is C_3 or $K_{2,3}$ then $|E(G)| = \left\lceil \frac{3}{2}|V(G)| - \frac{3}{2} \right\rceil$. We will now show that those two graphs are the unique critical ones satisfying this equality.

Farley and Proskurowski [6] constructed a class of multigraphs G called *ABC-graphs* with no matching-cut having $\left\lceil \frac{3}{2}|V(G)| - \frac{3}{2} \right\rceil$ edges. The definition of ABC-graphs is based on the following three operations:

- An *A-operation* on vertex u introduces vertices v and w and edges uv, uw and vw.
- A *B-operation* on edge uv introduces vertices w_1 and w_2 and edges uw_1, vw_1, uw_2 and vw_2, and removes edge uv.

- A *C-operation* on vertices u and v ($u = v$ is allowed) introduces vertex w and edges uw and vw.

Note that the C-operation is the only operation that can introduce parallel edges.

An *ABC-graph* is a graph that can be obtained from K_1 with a sequence of A- and B-operations and at most one C-operation.

It is easy to check that ABC-graphs have no matching-cut. In addition, solving a conjecture of Farley and Proskurowski [6], Bonsma [7, 10] showed that they are the unique extremal examples, i.e., satisfying $|E(G)| = \lceil \frac{3}{2}|V(G)| - \frac{3}{2} \rceil$.

Theorem 19 (Bonsma [7, 10]). *Let G be a graph such that $|E(G)| = \lceil \frac{3}{2}|V(G)| - \frac{3}{2} \rceil$. Then G has no matching-cut if and only if G is an ABC-graph.*

Our aim is to prove that every $\{C_3, K_{2,3}\}$-free ABC-graph is good. It is easy to check that every 2-connected component of an ABC-graph is an ABC-graph, so by Lemma 7, it suffices to prove it for 2-connected ABC-graphs.

Observe that the C-operation is the only one that changes the parity of the order. Hence an ABC-graph with an odd number of vertices is obtained from K_1 with a sequence of A- and B-operations and no C-operation.

Let G be a graph obtained from a graph H by a B-operation on some edge uv. Let ϕ be an edge-labelling of H. Let ϕ_0 and ϕ_∞ be the edge-labellings of G defined by:

$$\begin{aligned}
&\phi_0(e) = \phi_\infty(e) = \phi(e) \text{ for all } e \in E(H) \setminus \{uv\},\\
&\phi_0(uw_1) = \phi_0(w_2v) = 1/2,\\
&\phi_0(uw_2) = \phi_0(w_1v) = -1/2,\\
&\phi_\infty(uw_1) = \phi_\infty(w_2v) = +\infty,\\
&\phi_\infty(uw_2) = \phi_\infty(w_1v) = -\infty
\end{aligned}$$

Proposition 20. *Let G be a graph obtained from a graph H by a B-operation on some edge uv and ϕ be a good edge-labelling of H.*

(i) If ϕ is injective integer-valued and $\phi(uv) = 0$, then ϕ_0 is a good edge-labelling of G.

(ii) If ϕ is real-valued, then ϕ_∞ is a good edge-labelling of G.

Proof: (i) By contradiction, suppose that ϕ_0 is not a good edge-labelling of G. Then there exist two increasing independent (x, y)-paths P_1 and P_2 on G, for some $x, y \in V(G)$.

Since ϕ is a good edge-labelling of H, by the definition of ϕ_0 at least one edge of the set $E' = \{uw_1, uw_2, vw_1, vw_2\}$ belongs to some of the paths P_1 or P_2. Observe also that an increasing path in H cannot contain more than two edges of E'.

Suppose then that exactly one of the paths, say P_1, contains a non-empty intersection with the set E'. In this case, there would be two increasing paths in the edge-labelling ϕ of H. To prove this fact, let P_1' be the path obtained from P_1 by replacing the edges of the set $E' \cap E(P_1)$ by the edge uv. Observe that P_1' and P_2 would be two increasing paths of H under the edge-labelling ϕ, since $\phi(uv) = 0$.

Hence the paths P_1 and P_2 both contain some edge of the set E'. Suppose first that P_1 and P_2 contain exactly one edge of E' each. As P_1 and P_2 are independent, we assume that $uw_1 \in E(P_1)$ and $vw_1 \in E(P_2)$, without loss of generality. If $w_1 = y$, then the last edge of the (x,u)-subpath of P_1 has a label smaller than 0 (since ϕ is injective) and the same happens for the last edge of the (x,v)-subpath of P_2 (observe that at least one of these subpaths must be non-empty). Consequently, there would be two increasing paths (x,u)-paths or (x,v)-paths in H under the edge-labelling ϕ. Similarly, one may conclude that if $w_1 = x$, then there would also be two increasing paths on ϕ. It is just necessary to verify that the first edges of the (u,y)-subpath of P_1 and of the (v,y)-subpath of P_2 are greater than 0 (at least one of these edges exist) and that there would be two increasing (u,y)-paths or (v,y)-paths in H.

Finally, P_1 and P_2 cannot have both two edges from E' because they are independent.

(ii) The proof that ϕ_∞ is a good edge-labelling of G is similar to the proof of (i). In this case, P_1 and P_2 cannot contain just one edge of E'. Consequently, either $E(P_1) \subset E'$ or $E(P_2) \subset E'$. In any case, there would be and increasing (u,v)-path or an increasing (v,u)-path, which is a contradiction because there would be two increasing paths in H. □

Corollary 21. *If G is a graph obtained from a good graph by a B-operation, then G is good.*

Proof: It follows directly from Proposition 20. □

Lemma 22. *Let G be a 2-connected ABC-graph with an odd number of vertices. If $G \notin \{C_3, K_{2,3}\}$ then G is good.*

Proof: By contradiction, suppose that G is a counter-example to the statement. As every A-operation (with the exception of the transition $K_1 \to C_3$) creates a cut-vertex, by Lemma 7, we may assume that G is obtained from C_3 with a sequence of B-operations. However a B-operation on C_3 at any edge creates a $K_{2,3}$ and a B-operation on $K_{2,3}$ at any edge creates the graph G_1 depicted in Figure 4. If $G \notin \{C_3, K_{2,3}\}$ then it is obtained from G_1 with a sequence of B-operations. Now this graph G_1 admits a good edge-labelling (See Figure 4). Hence an easy induction and Corollary 21 imply that G has a good edge-labelling, a contradiction.

□

Since 2-connected components of an ABC-graph with an odd number of vertices are ABC-graphs with an odd number of vertices, we have the following:

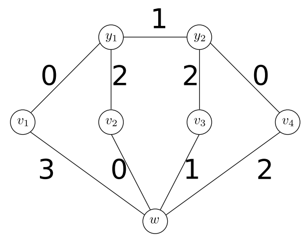


Figure 4: The graph G_1 and a good edge-labelling.

Corollary 23. *Every $\{C_3, K_{2,3}\}$-free ABC-graph with an odd number of vertices is good.*

We now would like to prove an analogous statement to the one of Corollary 23 but for ABC-graphs with an even number of vertices.

Let G be a graph and x, y be two distinct vertices of G. An (x, y)-*better edge-labelling* of G is a good edge-labelling of G such that there is no increasing (x, y)-path. Clearly, if x and y are adjacent or if x and y have two neighbours in common then G has no (x, y)-better edge-labelling. A graph is *friendly* if it has a good edge-labelling and for any pair (x, y) of non-adjacent vertices with at most one neighbour in common there exists an (x, y)-better edge-labelling.

Lemma 24. *G_1 is friendly.*

Proof: Let ϕ be the edge-labelling of G_1 in Figure 4. Then ϕ is good.

Let us now prove that for every pair $p = (a, b)$ of two distinct non-adjacent vertices a and b in G_1 such that a and b have at most one common neighbour and, there is a better (a, b)-edge-labelling of G_1.

First, observe that the vertex w of G_1 cannot be in such a pair because, for any other vertex of G_1, either w is adjacent to it or they have two common neighbours.

Suppose now that the vertex $y_1 \in p$. Then the other vertex of p must be v_3 or v_4. But ϕ is (v_3, y_1)-better and (y_1, v_4)-better, and so $-\phi$ is (y_1, v_3)-better and (v_4, y_1)-better. Hence in any case, there is a better p-edge-labelling of G_1.

By symmetry, if y_2 is a vertex of p, there exists a p-better edge-labelling.

Suppose that $v_1 \in p$. Then the other vertex of p is v_3 or v_4. ϕ is (v_1, v_4)-better and exchanging the labels of y_2v_3 and y_2v_4 and also the labels of v_3w and v_4w we obtain a (v_1, v_3)-better edge-labelling ϕ'. Thus $-\phi'$ and $-\phi$ are respectively (v_3, v_1)-better and (v_4, v_1)-better. Hence in any case, there is a better p-edge-labelling of G_1.

By symmetry, if v_2, v_3 or v_4 is a vertex of p, there exists a p-better edge-labelling. $\square$

Proposition 25. *Let G be a graph obtained from a graph H by a B-operation on some edge uv. If H is friendly then G is friendly.*

Proof: Let w_1, w_2 be the vertices created by the B-operation. Let x and y be two non-adjacent vertices of G having at most one neighbour in common. Then $|\{x,y\} \cap \{w_1,w_2\}| \leq 1$.

- Suppose first that $\{x,y\} \cap \{w_1,w_2\} = \emptyset$. Then x and y are not adjacent in H.

  Assume first that x and y have at most one common neighbour in H. Let ϕ be an injective integer-valued (x,y)-better edge-labelling of H such that $\phi(uv) = 0$. Then ϕ_0 is a good edge-labelling of G by Proposition 20-(i). Moreover it is easy to check that there is no increasing (x,y)-path in G. Hence ϕ_0 is an (x,y)-better edge-labelling of G.

  Assume now that x and y have two common neighbours in H. As they do not have two common neighbours in G, we can suppose w.l.o.g. that $x = u$ and $N(x) \cap N(y) = \{v,w\}$, for some vertex w. Let ϕ be a real-valued good edge-labelling of H. Free to consider $-\phi$, we may assume that uvy is an increasing path. Hence in $H \setminus uv$ there is no increasing (u,y)-path. By Proposition 20-(ii), ϕ_∞ is a good edge-labelling of G. Moreover it is an (x,y)-better edge-labelling, because there is no increasing (u,y)-path in $H \setminus uv$ and the unique increasing paths containing w_1 and w_2 are uw_2 and uw_2v.

- Suppose now that $|\{x,y\} \cap \{w_1,w_2\}| = 1$. Without loss of generality, we may assume that $x = w_1$ and y is not adjacent to v.

  Assume first that v and y have at most one common neighbour in H. Let ϕ be a (v,y)-better edge-labelling of H. By Proposition 2, we may assume that ϕ is real-valued. By Proposition 20-(ii), ϕ_∞ is a good edge-labelling of G. Moreover, there is no increasing (w_1,y)-path, through u since $\phi(uw_1) = +\infty$, nor through v since there is no increasing (v,y)-path in H. Hence ϕ_∞ is a (w_1,y)-better edge-labelling of G.

  Assume now that v and y have two common neighbours in H.

  – Suppose that y is adjacent to u. Let ϕ be an injective integer-valued good edge-labelling of H such that $\phi(uv) = 0$. Free to consider $-\phi$, we may assume that $\phi(uy) < 0$ and so $\phi(uy) \leq -1$. By Proposition 20-(i), ϕ_0 is a good edge-labelling of G. Moreover it has no increasing (w_1,y)-path and so is (w_1,y)-better. Indeed suppose for a contradiction that there is an increasing (w_1,y)-path P :

    ∗ If u is the second vertex of P then $P - w_1$ is an increasing (u,y)-path. Since $\phi(uy) \leq -1$, $P - w_1$ is not (u,y). So $P - w_1$ and (u,y) are two increasing (u,y)-paths in H a contradiction.

    ∗ If v is the second vertex of P then the path Q in H obtained from P by replacing w_1 with u is an increasing (u,y)-path because the labels of the edges of $P - w_1$ are positive. Thus Q and (u,y) are distinct increasing (u,y)-paths, a contradiction.

– Suppose that y is not adjacent to u. Let t_1 and t_2 be the two common neighbours of v and y. Let ϕ be an injective integer-valued good edge-labelling of H such that $\phi(uv) = 0$. Without loss of generality, we may assume that (v, t_1, y) is increasing and (v, t_2, y) is decreasing. By Observation 3.1, $\phi(vt_1) < \phi(vt_2)$. Thus, if $\phi(vt_1) > 0$ then $\phi(vt_2) > 0$. So with respect to $-\phi$, (v, t_2, y) is increasing and $-\phi(vt_2) < 0$. Hence, free to consider $-\phi$ (and swap the names of t_1 and t_2), we may assume that $\phi(vt_1) < 0$ and so $\phi(vt_1) \leq -1$. By Proposition 20-(i), ϕ_0 is a good edge-labelling of G. Moreover it has no increasing (w_1, y)-path and so is (w_1, y)-better. Indeed suppose for a contradiction that there is a increasing (w_1, y)-path P :

  * If v is the second vertex of P then $P - w_1$ is an increasing (v, y). Since $\phi(vt_1) \leq -1$, $P - w_1$ is not (v, t_1, y). So there are two increasing (v, y)-paths in H, a contradiction.
  * If u is the second vertex of P then the path P' in H obtained from P by replacing w_1 with v is an increasing (v, y)-path because the labels of the edges of $P - w_1$ are positive. P' is distinct from (v, t_1, y), a contradiction.

□

One can now generalize Lemma 22.

Lemma 26. *Let G be a 2-connected ABC-graph with an odd number of vertices. If $G \notin \{C_3, K_{2,3}\}$ then G is friendly.*

Proof: Similarly as in the proof of Lemma 22, combining Lemma 24 and Proposition 25 yield the result by induction. □

Corollary 27. *Every $\{C_3, K_{2,3}\}$-free ABC-graph with an odd number of vertices is friendly.*

Proof: Let x and y be two non-adjacent vertices of G having at most one common neighbour.

Assume first that x and y are in a same connected 2-component C. By Lemma 26, C has an (x, y)-better edge-labelling and, by Corollary 23, $G \setminus E(C)$ has a good edge-labelling. The union of these two edge-labellings is clearly an (x, y)-better labelling of G.

Suppose now that the 2-connected components containing x do not contain y. Let G_1 be the graph induced by the union of the 2-connected components containing x and $G_2 = G \setminus E(G_1)$. By Corollary 23, the two graphs G_1 and G_2 admit good edge-labellings ϕ_1 and ϕ_2, respectively. Free to add a huge number to all the labels of ϕ_1, we may assume that $\min\{\phi_1(e) \mid e \in E(G_1)\} > \max\{\phi_2(e) \mid e \in E(G_2)\}$. Then the union of ϕ_1 and ϕ_2 is an (x, y)-better labelling of G. □

Lemma 28. *Let G be a 2-connected ABC-graph with an even number of vertices. If G is $\{C_3, K_{2,3}\}$-free, then G is good.*

Proof: We prove this lemma by induction on the number of vertices (or equivalently the number of A-, B- or C-operations). An even ABC-graph is obtained from K_1 with a sequence of A- and B-operations and exactly one C-operation. Since G is 2-connected, no A-operation can be made after a C-operation. Consider a sequence of operations such that the C-operation is done as late as possible. Let u and v be the vertices on which the C-operation is done and w the introduced vertex.

- Suppose that the C-operation is the ultimate one. Note that $u \neq v$ since G has no multiple edges. Since G is $\{C_3, K_{2,3}\}$-free then u and v are not adjacent and u and v have at most one neighbour in common. Hence by Corollary 27, $G - w$ admits a (u, v)-better edge-labelling ϕ (in $\mathbb{R}$). Setting $\phi(uw) = -\infty$ and $\phi(wv) = +\infty$ we obtain a good edge-labelling of G.

- If the C-operation is the penultimate one, then it is followed by a B-operation on one of the introduced edges, because the C-operation is applied as late as possible and G is C_3-free. These two operations together may be seen as a single one on u and v that introduces the vertices t_1, t_2 and w and the edges ut_1, ut_2, t_1w, t_2w and wv.

  Note that u and v are not adjacent since G is $K_{2,3}$-free. Assume first that u and v have at most one neighbour in common. By Corollary 27, $G - \{t_1, t_2, w\}$ admits a (u, v)-better edge-labelling ϕ. Let M be the maximum value of ϕ. Then setting $\phi(ut_1) = \phi(t_2w) = -\infty$, $\phi(ut_2) = \phi(t_1w) = M+1$ and $\phi(vw) = M + 2$, we obtain a good edge-labelling of G.

  Assume now that u and v have at least two common neighbours. Since G is $K_{2,3}$-free, then u and v have exactly two common neighbours x_1 and x_2. By Corollary 23, $G - \{t_1, t_2, w\}$ admits a good edge-labelling ϕ. By Proposition 2, we may assume that ϕ is injective and real-valued. Without loss of generality, we may suppose that $\phi(vx_1) > \phi(vx_2)$. Let us set $\phi(ut_1) = \phi(t_2w) = +\infty$, $\phi(ut_2) = \phi(t_1w) = -\infty$ and $\phi(vw) = \frac{1}{2}(\phi(vx_1)+\phi(vx_2))$. We claim that ϕ is a good edge-labelling of G. Indeed suppose, by way of contradiction, that it is not the case. Then there exist two vertices a and b and two independent increasing (a, b)-paths P_1 and P_2. Since ϕ is a good edge-labelling of $G - \{t_1, t_2, w\}$ one of these two paths, say P_1 must go through w. Moreover since $\phi(t_1w) = -\infty$ and $\phi(t_2w) = +\infty$ and $d(w) = 3$, then either wt_1 (or t_1w) is the first edge of P_1 or t_2w (or wt_2) is the last edge of P_1. Free to consider $-\phi$ instead of ϕ, we may assume that we are in the first case.

  Two cases may occur. Either (a) P_1 starts in t_1 or (b) P_1 starts in w.

  (a) In this case, $P_2 = (t_1, u)$ and the third vertex of P_1 is v. Then $Q_1 = P_1 - \{t_1, w\}$ is an increasing (v, u)-path. So by Observation 3.1 and the assumption that $\phi(vx_1) > \phi(vx_2)$, $Q_1 = vx_2u$ (We recall the reader that another increasing (v, u)-path not going through x_2 cannot exist as ϕ is a good edge-labelling of $G - \{t_1, t_2, w\})$. This is a contradiction because $\phi(wv) > \phi(vx_2)$.

(b) In this case, $P_1 = (w, t_1, u)$, because $\phi(ut_1) = +\infty$. Now the first edge of P_2 is wv. Hence $Q_2 = P_2 - w$ is an increasing (v, u)-path and vx_2 is not the first edge of Q_2 since $\phi(wv) > \phi(vx_2)$. Note that by Observation 3.1, vx_2u is increasing because $\phi(vx_1) > \phi(vx_2)$. So, in $G - \{t_1, t_2, w\}$, there are two distinct increasing (v, u)-paths. This contradicts the fact that ϕ is a good edge-labelling of $G - \{t_1, t_2, w\}$.

- If there are exactly two B-operations after the C-operation, and if u and v are not adjacent then by the induction hypothesis and Corollary 21, G has a good edge-labelling. If u and v are adjacent, then uv is a K_2-cut. Let C_1 be the component of $G - \{u, v\}$ containing w (i.e., the set of vertices added with the C-operation and the following B-operations). Let $G_1 = G\langle C_1 \cup \{u, v\}\rangle$ and $G_2 = G\langle V(G) \setminus C_1\rangle$. Note that G_1 is obtained from a triangle by performing two B-operations and thus is the graph G_1 depicted Figure 4 which has a good edge-labelling. Similarly, G_2 is the graph G taken before performing the C-operation has a good edge-labelling. Hence by Lemma 10, G has a good edge-labelling.
- If there are at least three B-operations after the C-operation, then by the induction hypothesis and Corollary 21, G has a good edge-labelling.

$\square$

Lemma 22 and Lemma 28 imply that every 2-connected $\{C_3, K_{2,3}\}$-free ABC-graph is good. Since 2-connected components of an ABC-graph are ABC-graphs, we have the following.

Corollary 29. *Every $\{C_3, K_{2,3}\}$-free ABC-graph is good.*

In turn, this corollary, together with Corollary 12, Theorems 16 and 19, yield the following.

Theorem 30. *Let G be a critical graph. If $G \notin \{C_3, K_{2,3}\}$ then $|E(G)| \geq \frac{3}{2}|V(G)| - \frac{1}{2}$.*

7. Conclusions and further research

We have shown that it is NP-complete to decide if a graph has a good edge-labelling, even for the class of bipartite graphs. It would be nice to find large classes of graphs for which it is polynomial-time decidable. For graphs with treewidth 1, which are the forests, it is the case. But is it also the case for graphs with treewidth at most k?

Problem 31. *Let $k \geq 2$ be a fixed integer. Does there exist a polynomial-time algorithm that decides if a given graph of treewidth at most k has a good edge-labelling?*

We also do not know what is the complexity of the problem when restricted to planar graphs.

Problem 32. *Does there exist a polynomial-time algorithm that decides if a given planar graph has a good edge-labelling?*

We do not even know if there are planar critical graphs distinct from C_3 and $K_{2,3}$.

Problem 33. *Does there exist a* $\{C_3, K_{2,3}\}$*-free planar graph which is bad?*

If there is no such graphs or only a finite number of them then the answer to Problem 32 will be yes.

Corollary 18 implies that, with the additional condition of girth at least 6, the answer to Problem 33 is no. It would be nice to solve the above problems for planar graphs of smaller girth. In particular, we do not know if there is a planar graph with girth 5 which is bad.

Problem 34. Does every planar graph of girth at least 5 have a good edge-labelling?

Bonsma [11] showed that it is NP-complete to decide if a planar graph of girth at least 5 has a matching-cut. In particular, there are infinitely many planar graphs of girth at least 5 without matching-cut. However, for all such graphs we looked at, we were able to find a good edge-labelling.

. The *average degree* of a graph G is $Ad(G) = \frac{\sum_{v \in V(G)} d(v)}{|V(G)|} = \frac{2|E(G)|}{|V(G)|}$.

Theorem 30 implies that for any $c < 3$ there is a finite number of critical graphs with average degree at most c. Actually, we conjecture that the only ones are C_3 and $K_{2,3}$.

Conjecture 35. *Let G be a critical graph. Then $Ad(G) \geq 3$ unless $G \in \{C_3, K_{2,3}\}$.*

More generally for any $c < 4$, we conjecture the following.

Conjecture 36. *For any $c < 4$, there exists a finite list of graphs $\mathcal{L}$ such that if G is a critical graph with $Ad(G) \leq c$ then $G \in \mathcal{L}$.*

The constant 4 in the above conjecture would be tight. Indeed, for all k, the graph $J_{2,2,k}$ defined in Section 3 is critical: it is bad according to Proposition 4. Moreover one can easily show that for any edge e, $H_k \setminus e$ has a good edge-labelling with no (u,v)-increasing path and no (v,u)-increasing (just follow the constraint as in the proof of Proposition 3). Extending this labelling by labelling the two H_2 with $-\infty$ and $+\infty$ such that one of them is an increasing (u,v)-path and the other one an increasing (v,u)-path we obtain a good edge-labelling of $J_{2,2,k} \setminus e$. Furthermore $Ad(J_{2,2,k}) = \frac{8k+8}{2k+4} = 4 - \frac{4}{k+2}$. Last, one can easily see that if $k \neq k'$ then $J_{2,2,k}$ is not a subgraph of $J_{2,2,k'}$.

. Theorem 30 says that if a graph has no dense subgraphs then it has a good edge-labelling. On the opposite direction one may wonder what is the minimum density ensuring a graph to be bad. Or equivalently,

Problem 37. What is the maximum number $g(n)$ of edges of a good graph on n vertices?

Clearly we have $g(n) = ex(n, \mathcal{C})$ where $\mathcal{C}$ is the set of critical graphs. As $K_{2,3}$ is critical then $g(n) \leq ex(n, K_{2,3}) = \frac{1}{\sqrt{2}} n^{3/2} + O(n^{4/3})$ by a result of Füredi [12].

The hypercubes show that g is super-linear. Indeed the hypercube H_k is obtained from two disjoints copies of H_{k-1} by adding a perfect matching between them. Hence an easy induction and Lemma 11 shows that H_k has a good edge-labelling. Since H_k has 2^k vertices and $2^{k-1}k$ edges, $g(2^k) \geq 2^{k-1}k$, so $g(n) \geq \frac{1}{2} n \log n$.

References

[1] J.-C. Bermond, M. Cosnard, S. Pérennes, Directed acyclic graphs with unique path property, Tech. Rep. RR-6932, INRIA (May 2009).

[2] B. Mukherjee, Optical communication networks, McGraw-Hill New York, 1997.

[3] R. Ramaswami, K. Sivarajan, Routing and wavelength assignment in all-optical networks, IEEE/ACM Transactions on Networking (TON) 3 (5) (1995) 489–500.

[4] K. Bala, T. Stern, Algorithms for routing in a linear lightwave network, in: IEEE INFOCOM'91. Proceedings. Tenth Annual Joint Conference of the IEEE Computer and Communications Societies. Networking in the 90s., 1991, pp. 1–9.

[5] J.-S. Sereni, Private communication.

[6] A. Farley, A. Proskurowski, Extremal graphs with no disconnecting matching, in: Proceedings of the second West Coast conference on combinatorics, graph theory, and computing, 1984, pp. 153–165.

[7] P. Bonsma, A characterization of extremal graphs with no matching-cut, in: Proceedings of the European conference on combinatorics, graph theory and applications (EuroComb 05), Discrete Mathematics and Theoretical Computer Science Proceedings AE, Nancy, 2005, pp. 135–138. URL `http://www.dmtcs.org/proceedings/abstracts/dmAE0127.abs.html`

[8] T. J. Schaefer, The complexity of satisfiability problems, in: STOC '78: Proceedings of the tenth annual ACM symposium on Theory of computing, ACM, New York, NY, USA, 1978, pp. 216–226. doi:http://doi.acm.org/10.1145/800133.804350.

[9] N. Eaton, T. Hull, Defective list colorings of planar graphs, Bull. Inst. Combin. Appl 25 (1999) 79–87.

[10] P. Bonsma, A characterization of extremal graphs with no matching-cut, Tech. Rep. Memorandum 1769, University of Twente (2005).
URL `http://www.math.utwente.nl/publications/2005/1769abs.html`

[11] P. Bonsma, The complexity of the matching-cut problem for planar graphs and other graph classes, J. Graph Theory 62 (2) (2009) 109–126.

[12] Z. Füredi, New asymptotics for bipartite Turán numbers, J. Combin. Theory Ser. A 75 (1) (1996) 141–144.

Graph Classes (Dis)satisfying the Zagreb Indices Inequality

Vesna Andova,[a1] **Nathann Cohen**[b] and **Riste Škrekovski**[c]

[a]*Institute of Mathematics and Physics, Faculty of Electrical Engineering and Information Technologies, Ss Cyril and Methodius Univ., Ruger Boskovik, P. O. Box 574, 1000 Skopje, Macedonia*
e-mail: vesna.andova@gmail.com

[b]*Projet Mascotte, I3S(CNRS, UNSA) and INRIA, 2004 route des lucioles, BP 93, 06902 Sophia–Antipolis Cedex, France*
e-mail: nathann.cohen@gmail.com

[c]*Institute of Mathematics, Physics, and Mechanics, Jadranska 19, 1111 Ljubljana, Slovenia*
e-mail: skrekovski@gmail.com

(Received January 22, 2010)

Abstract

Recently Hansen and Vukičević [11] proved that the inequality $M_1/n \leq M_2/m$, where M_1 and M_2 are the first and second Zagreb indices, holds for chemical graphs, and Vukičević and Graovac [23] proved that this also holds for trees. In both works a distinct counterexample is given for which this inequality is false in general. Here, we present some classes of graphs with prescribed degrees, that satisfy $M_1/n \leq M_2/m$. Namely every graph G whose degrees of vertices are in the interval $[c, c+\lceil\sqrt{c}\,\rceil]$ for some integer c, satisfies this inequality. In addition, we prove that for any $\Delta \geq 5$, there is an infinite family of connected graphs of maximum degree Δ, such that the inequality is false.

1 Introduction

The first and second Zagreb indices are among the oldest topological indices [2, 8, 10, 14, 21], defined in 1972 by Gutman and Trinajstić [9], and are given different names in the literature,

[1]Corresponding author.

such as the Zagreb group indices, the Zagreb group parameters and most often, the Zagreb indices. Zagreb indices were among the first indices introduced, and have since been used to study molecular complexity, chirality, ZE-isomerism and hetero-systems. Overall, Zagreb indices exhibit a potential applicability for deriving multi-linear regression models. The article [18] was responsible for a new research wave concerning Zagreb indices.

In the following, let $G = (V, E)$ be a simple graph with $n = |V|$ vertices and $m = |E|$ edges. These indices are defined as

$$M_1(G) = \sum_{v \in V} d(v)^2 \quad \text{and} \quad M_2(G) = \sum_{uv \in E} d(u)d(v) \ .$$

For the sake of simplicity, we often use M_1 and M_2 instead of $M_1(G)$ and $M_2(G)$, respectively. See [6, 7, 16, 12, 26, 27, 28] for more work done on these indices. Comparing the values of these indices on the same graph gives interesting results. At first the next conjecture was proposed [1, 3, 4]:

Conjecture 1.1. *For all simple graphs* G *,*

$$\frac{M_1(G)}{n} \leq \frac{M_2(G)}{m} \tag{1}$$

and the bound is tight for complete graphs.

One can see that this relation becomes an equality on regular graphs, but also when G is a star. Besides, the inequality is true for trees [23], graphs of maximum degree four, so called chemical graphs [11] and unicyclic graphs [25], even though it does not hold for general graphs. See [11, 23, 5, 13, 20] for various examples of graphs dissatisfying the inequality (1).

In this article, we present some other classes of graphs with prescribed degrees for which (1) holds, and more generally conditions on the distribution of degrees in a graph G implying the relation (1). We also show that there are arbitrarily long intervals $[a, b]$ such that a graph with minimal degree at least a and maximum degree at most b satisfies the same relation. Namely, every graph G, such that its vertex degrees are in the interval $[c, c + \lceil \sqrt{c}\,\rceil]$ for any integer c, satisfies this inequality. We also prove that for any $\Delta \geq 5$, there is an infinite family of connected graphs of maximum degree Δ such that the inequality is false.

We denote by $K_{a,b}$ the *complete bipartite* graph with a vertices in one class and b vertices in the other one. We call *k-star* the star on k edges, and *k-path* the path of length k. Since we discuss necessary conditions for (1) to hold, we denote for the sake of simplicity by $m_{i,j}$ the number of edges that connect vertices of degrees i and j in the graph G. Then, as shown in [11]:

$$\frac{M_2}{m} - \frac{M_1}{n} = \sum_{\substack{i\leq j\\ k\leq l\\ (i,j),(k,l)\in\mathbb{N}^2}} \left[\left(i\,j\,\left(\frac{1}{k}+\frac{1}{l}\right)+k\,l\,\left(\frac{1}{i}+\frac{1}{j}\right)-i-j-k-l\right)\,m_{i,j}\,m_{k,l}\right]. \quad (2)$$

Sometimes in order to examine whether the inequality (1) holds, one can consider whether $M_2/m - M_1/n$ is non-negative. The difference that we are considering is given by (2). In order to simplify (2), we will define a function f, and study some of its properties. Now, for integers i, j, k, l, let

$$f(i,j,k,l) = i\,j\,\left(\frac{1}{k}+\frac{1}{l}\right)+k\,l\,\left(\frac{1}{i}+\frac{1}{j}\right)-i-j-k-l\ .$$

Then (2) can be restated as

$$\frac{M_2}{m} - \frac{M_1}{n} = \sum_{\substack{i\leq j,k\leq l\\ (i,j),(k,l)\in\mathbb{N}^2}} f(i,j,k,l)m_{i,j}\,m_{k,l}\ . \quad (3)$$

2 Some properties of f

In the sequel, we study some properties of the function f.

Lemma 2.1. *For any integers i, j, k, l, it holds $f(i,j,k,l)<0$ if and only if*

(a) $ij > kl$ *and* $\frac{1}{k}+\frac{1}{l} < \frac{1}{i}+\frac{1}{j}$ *or*

(b) $ij < kl$ *and* $\frac{1}{k}+\frac{1}{l} > \frac{1}{i}+\frac{1}{j}$.

Proof. This result follows immediately by the decomposition of f. Namely

$$\begin{aligned} f(i,j,k,l) &= \frac{ij}{kl}(k+l)-(k+l)+\frac{kl}{ij}(i+j)-(i+j) \\ &= (k+l)\left[\frac{ij-kl}{kl}\right]+(i+j)\left[\frac{kl-ij}{ij}\right] \\ &= (ij-kl)\left(\frac{1}{k}+\frac{1}{l}-\frac{1}{i}-\frac{1}{j}\right). \end{aligned}$$

□

Notice that the function f has some symmetry properties, namely for every i, j, k and l:

$$f(i,j,k,l) = f(j,i,k,l) \qquad \text{and} \qquad f(i,j,k,l) = f(k,l,i,j)\ .$$

Determining the sign of the function f will help us to see whether the difference $M_2/m - M_1/n$ is non-negative. The following lemma gives us orderings of the integers i, j, k, and l, for which $f(i,j,k,l)$ can be negative.

Lemma 2.2. *If $f(i,j,k,l) < 0$ for some integers $i \le j$ and $k \le l$, then*

$$i < k \le l < j \qquad \textit{or} \qquad k < i \le j < l\ .$$

Proof. Suppose first that $i \le k$. There are only three possibilities:

- $i \le j \le k \le l$;
- $i \le k \le j \le l$;
- $i \le k \le l \le j$.

If $i \le j \le k \le l$, then $ij \le kl$, but $\frac{1}{k} + \frac{1}{l} < \frac{1}{i} + \frac{1}{j}$, so this is impossible by Lemma 2.1(a). If $i \le k \le j \le l$, then $ij \le kl$ and $\frac{1}{k} + \frac{1}{l} < \frac{1}{i} + \frac{1}{j}$. This ordering is also impossible by Lemma 2.1(a). So, the only possible ordering for $f(i,j,k,l)$ to be negative is $i \le k \le l \le j$.

Now, if $i = k\,(i = k \le l \le j)$, then $ij \ge kl$ and $\frac{1}{i} + \frac{1}{j} > \frac{1}{k} + \frac{1}{l}$, which contradicts Lemma 2.1 (a). So, we conclude that $i < k$. Similarly, one can show that $l \ne j$. Thus, we obtain the first ordering $i < k \le l < j$ given in the lemma.

Suppose now that $k \le i$. Applying a similar argument as above, one obtains that $k < i \le j < l$ is the only possible ordering. □

3 Small good sets

It is easy to see that if G is a k-regular graph, then (1) is valid, since

$$\frac{M_1}{n} = k^2 = \frac{M_2}{m}\ .$$

As Conjecture 1.1 is false in general, but true for k-regular graphs, one may wonder if it also holds for "almost regular" graphs, i.e., graphs with only few vertex degrees. Now, we verify that this holds for graphs with only two vertex degrees. We give a direct short proof avoiding using the properties of the function f.

Proposition 3.1. *Let $x, y \in \mathbb{N}$, and let G be a graph with n vertices, m edges, and $d(v) \in \{x, y\}$ for every vertex v of G. Then, the inequality* (1) *holds for G.*

Proof. Since $d(v) = x$ or y for every vertex $v \in V$, we conclude that $m_{i,j} = 0$, whenever $i, j \notin \{x, y\}$. By (2), we infer

$$\begin{aligned}\frac{M_2}{m} - \frac{M_1}{n} &= 2\left[\frac{x^3(x-y)^2}{x^3 y} m_{x,x}\, m_{x,y} + \frac{2xy(x-y)^2(x+y)}{x^2 y^2} m_{x,x}\, m_{y,y}\right. \\ &+ \left.\frac{y^3(x-y)^2}{xy^3} m_{x,y}\, m_{y,y}\right] \\ &= 2(x-y)^2\left[\frac{1}{y} m_{x,x}\, m_{x,y} + 2\left(\frac{1}{x} + \frac{1}{y}\right) m_{x,x}\, m_{y,y} + \frac{1}{x} m_{x,y}\, m_{y,y}\right] \\ &\geq 0\end{aligned}$$

which establishes the claim. $\square$

Let $D(G)$ be the set of the vertex degrees of G, i.e., $D(G) = \{d(v) \mid v \in V\}$. Motivated by the above proposition, one may be interested to look for the sets D with property that for every graph G with $D(G) \subseteq D$ the inequality (1) holds. Hence, it is reasonable to introduce the following definition: A set S of integers is *good* if for every graph G with $D(G) \subseteq S$, the inequality (1) holds. Otherwise, S is a *bad* set. Thus, by above any set of integers of size ≤ 2 is good.

In Proposition 3.1 we have shown that for a graph G with $|D(G)| = 2$, the inequality (1) holds. Sun and Chen [19] showed that any graph G with $\Delta(G) - \delta(G) \leq 2$ satisfies (1). Thus, any interval of length three is good. One can generalize this result in the following way:

Proposition 3.2. *Let $s, x \in \mathbb{N}$. For every graph G with n vertices, m edges, and $D(G) \subseteq \{x - s, x, x + s\}$, the inequality (1) holds.*

Proof. The inequality (1) holds if $M_2/m - M_1/n$ is non-negative. The difference (3) is non-negative if for any integers i, j, k, l, the function $f(i, j, k, l)$ is non-negative. So we are interested whether $f(i, j, k, l)$ can be negative for some integers i, j, k, l. By Lemma 2.2, we may assume, up to symmetry, that the ordering of i, j, k, l is $i < k \leq l < j$. Since

$i,j,k,l \in \{x-s, x, x+s\}$, we have that $f(i,j,k,l)$, can be negative only if $i = x-s, k = l = x$ and $j = x+s$. But $f(x-s, x+s, x, x) = \dfrac{1}{x-s} - \dfrac{2}{x} + \dfrac{1}{x+s} > 0$. Hence, we conclude that

$$\frac{M_2}{m} - \frac{M_1}{n} = \sum_{\substack{i \le j, k \le l \\ (i,j),(k,l) \in \mathbb{N}^2}} f(i,j,k,l)\, m_{i,j}\, m_{k,l} > 0 \, . \qquad \square$$

Notice that the above result cannot be extended to any interval of length 4 as Sun and Chen [19] gave a non-connected counterexample. For connected one, consider the graph $G(l,k,s)$ with $l = 4$ from Fig. 2. It is obvious that $D\left(G(4,k,s)\right)$ is a subset of the interval $[2,5]$, but this graph for proper values of k and s does not satisfy the inequality (1), see Theorem 5.1. Both graphs contain vertices of degree 2. It is interesting that Sun and Chen [19] proved that any graph G with $\Delta(G) - \delta(G) \le 3$ and $\delta(G) \ne 2$ satisfy (1). Thus, any interval $[x, x+3]$ is good with only exception of $[2,5]$.

The proof of Proposition 3.2 motivates a more general conclusion.

Proposition 3.3. *The set of integers* $\{a,b,c\}$ *, where* $a < b < c$ *, is good if and only if*

(*a*) $b^2 \ge ac$ *and* $b(a+c) \ge 2ac$ *, or*

(*b*) $b^2 \le ac$ *and* $b(a+c) \le 2ac$ *.*

Proof. Since $a < b < c$, by Lemma 2.2 the function f can be negative in $f(i,j,k,l)$ only if either $i = a$, $k = l = b$ and $j = c$, or $k = a$, $i = j = b$ and $l = c$, i.e., only $f(a,c,b,b) = f(b,b,a,c) = (ac - b^2)\left(\dfrac{2}{b} - \dfrac{1}{a} - \dfrac{1}{c}\right)$ can be negative. If (a) or (b) holds, then it is obvious that $f(i,j,k,l) \ge 0$ for any integers $i,j,k,l \in \{a,b,c\}$, and the inequality (1) is valid for every graph G such that $D(G) = \{a,b,c\}$.

For the other direction, suppose that neither (a) nor (b) holds. If this is the case, then only $f(a,c,b,b) < 0$. We construct a graph $G_{x,y}$ with $D(G_{x,y}) = \{a,b,c\}$, $m_{a,a} = m_{c,c} = 0$ and $m_{a,b} = m_{b,c} = 1$ (see Fig. 1). The graph $G_{x,y}$ can be created in the following way:

- Make a sequence of x copies of $K_{a,c}$ and then continue that sequence with y copies of $K_{b,b}$.
- Choose an edge from the first $K_{a,c}$ graph and another edge from the second $K_{a,c}$. Then replace these edges by edges connecting the "a"-vertex from the first graph with "c"-vertex from the second graph, and another edge connecting the "c"-vertex from

the first graph with "a"-vertex from the second graph. This way the degrees of the vertices are not changed. Continue this procedure between all x copies of $K_{a,c}$.

- Next, chose an edge from the last $K_{a,c}$ in the sequence and one edge from the first $K_{b,b}$ graph, replace these edges by edges connecting the "a"-vertex with one of the "b" vertices and the "c"-vertex with the other "b" vertex.

- The same procedure is applied between all consecutive graphs $K_{b,b}$ in the sequence and this way is $G_{x,y}$ constructed.

We emphasize that this binding procedure is done only once between $K_{a,c}$ and $K_{b,b}$ graphs.

Now,

$$\begin{aligned}
\frac{M_2}{m} - \frac{M_1}{n} &= \sum_{\substack{i\le j,k\le l\\ i,j,k,l\in\{a,b,c\}}} f(i,j,k,l) m_{i,j}\, m_{k,l} \\
&= 2\left[f(a,c,b,b) m_{a,c} m_{b,b} + \left[f(a,c,a,b) + f(a,c,b,c)\right] m_{a,c}\right. \\
&\quad \left. + \left[f(a,b,b,b) + f(c,b,b,b)\right] m_{b,b} + f(a,b,b,c)\right] .
\end{aligned}$$

If we increase the number of $K_{a,c}$ and $K_{b,b}$ graphs, i.e., x and y , in the graph $G_{x,y}$, shown on Fig. 1, then $m_{a,c}$ and $m_{b,b}$ will increase as well. For $m_{a,c}$ and $m_{b,b}$ big enough, the difference $M_2/m - M_1/n$ will be negative.

□

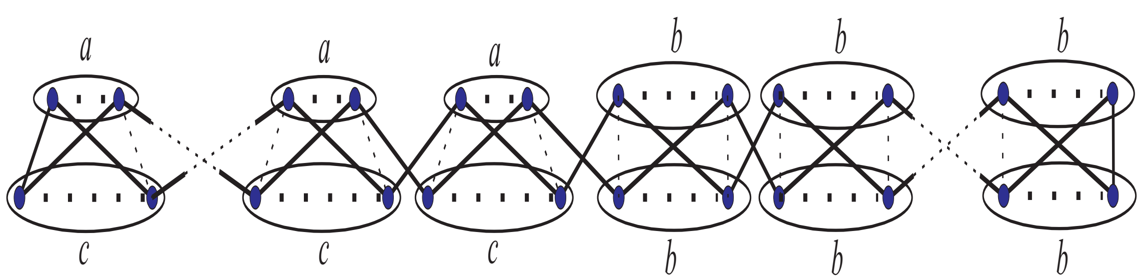


Figure 1: A connected graph G with $D(G) = \{a, b, c\}$. The edges that should be removed are drown with dashed lines.

4 Long good intervals

Our next goal is to determine long good intervals. We will need the following lemma to prove Theorem 4.1.

Lemma 4.1. *For integers c, i, j, and $p \le \lceil \sqrt{c}\,\rceil$ holds:*

$$c(c+p) > (c+i)(c+j) \quad \textit{if and only if} \quad i+j<p\,.$$

Proof. First notice that $ij \le \dfrac{(i+j)^2}{4}$. If $c(c+p) > (c+i)(c+j)$ and $i+j \ge p$, then

$$\begin{aligned} c^2 + cp &> c^2 + (i+j)c + ij \\ cp &> (i+j)c + ij, \end{aligned}$$

which is impossible. For the other direction, suppose that $i+j<p$. Then

$$\begin{aligned} (c+i)(c+j) &= c^2 + (i+j)c + ij \\ &\le c^2 + c(i+j) + \frac{(i+j)^2}{4} \\ &\le c^2 + c(p-1) + \frac{(p-1)^2}{4} \\ &< c(c+p) - c + \frac{(\sqrt{c})^2}{4} \\ &< c(c+p). \end{aligned}$$

This argument completes the proof. □

Using the previous lemma we can construct good intervals of any size. Notice that the following result holds for $c \le 9$ by the results of Sun and Chen [19] mentioned in the previous section, as in these cases the considered interval is of length at most 4.

Theorem 4.1. *For every integer c, the interval $[c, c+\lceil\sqrt{c}\,\rceil]$ is good.*

Proof. In order to prove the theorem, it is enough to show that $f(i,j,k,l) \ge 0$ whenever $i,\ j,\ k,\ l \in [c, c+\lceil\sqrt{c}\,\rceil]$. Suppose in contrary that for some i, j, k, l from this interval $f(i,j,k,l) < 0$. By Lemma 2.2, without loss of generality we can assume that $i<k\le l<j$. Now, let $k=i+s$, $l=i+t$, $j=i+q$ where $0<s\le t<q\le\lceil\sqrt{c}\,\rceil$. Now

$$\frac{1}{k}+\frac{1}{l}=\frac{2i+s+t}{(i+s)(i+t)} \qquad \text{and} \qquad \frac{1}{i}+\frac{1}{j}=\frac{2i+q}{i(i+q)}\,.$$

If $ij>kl$, then by Lemma 4.1 $s+t<q$. Hence $st<\dfrac{q^2}{4}$. By Lemma 2.1, $f(i,j,k,l)<0$, if $\frac{1}{k}+\frac{1}{l}<\frac{1}{i}+\frac{1}{j}$. Hence

$$\begin{aligned}
\frac{2i+s+t}{(i+s)(i+t)} &< \frac{2i+q}{i(i+q)} \\
(2i+s+t)(i^2+iq) &< (2i+q)(i^2+(s+t)i+st) \\
2i^3+(s+t+2q)i^2+(s+t)iq &< 2i^3+(2s+2t+q)i^2+2sti+(s+t)iq+stq \\
i^2\,q &< (s+t)\,i^2+2s\,t\,i+s\,t\,q \\
i^2\,q &< (q-1)\,i^2+2s\,t\,i+s\,t\,q
\end{aligned}$$

from here

$$\begin{aligned}
i^2 &< 2s\,t\,i+s\,t\,q \\
&< 2\frac{q^2}{4}\,i+\frac{q^3}{4}
\end{aligned}$$

which is clearly impossible since $q\le\lceil\sqrt{i}\,\rceil$.

Similarly, if $ij<kl$, then $s+t\ge q$. The function f in $f(i,j,k,l)$ is negative if and only if $\frac{1}{i}+\frac{1}{j}>\frac{1}{k}+\frac{1}{l}$. The last inequality implies

$$\begin{aligned}
i^2\,q &> (s+t)\,i^2+2s\,t\,i+s\,t\,q \\
&\ge q\,i^2+2s\,t\,i+s\,t\,q
\end{aligned}$$

and obviously this is impossible.

So $f(i,j,k,l)\ge 0$, for arbitrary i,j,k,l from the interval $[c,c+\lceil\sqrt{c}\,\rceil]$. $\square$

Theorem 4.1 is best in the sense that for $c=2$ the interval $[2,4]$ is good, but the interval $[2,5]$ is not. The following corollaries are immediate consequences of the above theorem.

Corollary 4.1. *If G is a graph with $\Delta(G)-\delta(G)\le\lceil\sqrt{c}\,\rceil$ and $\delta(G)\ge c$ for some integer c , then G satisfies the inequality (1).*

Corollary 4.2. *There are arbitrary long good intervals.*

5 Graphs of maximum degree at least 5

As we already mentioned, the inequality (1) holds for chemical graphs, but not in general. In [11, 23, 5, 13, 20, 19], examples of connected simple graph G are given such that $M_1/n > M_2/m$. What strikes the eye in these counterexamples is that either the maximum vertex degree is at least 10 or the graph is disconnected. We now produce for each $\Delta \geq 5$ an infinite family of connected planar counterexamples to (1) of maximum degree Δ.

Theorem 5.1. *There exists infinitely many graphs G of maximum degree ≥ 5 for which*

$$\frac{M_1}{n} > \frac{M_2}{m} \ .$$

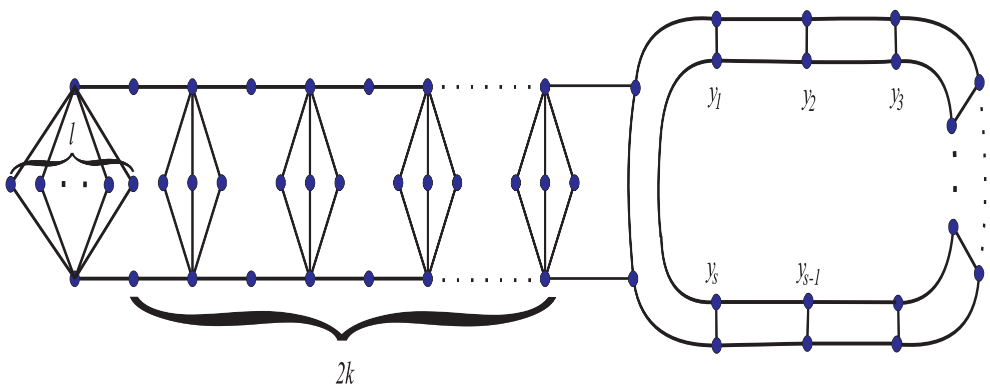


Figure 2: $G(l, k, s)$

Proof. Let G be the graph shown on the Fig. 2. This graph has $2k$ vertices of degree 5, $2s+2$ of degree 3, $5k+l$ vertices of degree 2 and two vertices of degree $l+1$. Also $m_{5,2} = 10k-2$, $m_{3,3} = 3s+2$, $m_{3,5} = 2$ and $m_{l+1,2} = 2(l+1)$. Then $n = 7k+2s+l+4$, $m = 10k+3s+2l+4$, $M_1 = 2(35k+9s+l^2+4l+10)$, $M_2 = 100k+27s+4l^2+8l+32$. From here one can obtained that

$$mM_1 - nM_2 = -2l^2 s + k(-144+64l-8l^2+s) - 8(6+5s) + l(8+17s) \ .$$

For every l, we can find k and s big enough such that $mM_1 - nM_2 > 0$. Obviously, we can find infinitely many such pair (k, s). $\square$

Observe that the right side of the graph $G(l, k, s)$ is the cubic graph $K_2 \square C_s$ with one edge twice subdivided. This graph can be substituted with any other cubic graph of appropriate size. $G(4,9,33)$ is the smallest graph for which the inequality of Theorem 5.1 holds, and it has 137 vertices.

Acknowledgement. This work was partially supported by the bilateral project BI-FR/09-10-PROTEUS-008. N. C. thanks for support by the ANR Blanc AGAPE and ANR Blanc International–Taiwan GRATEL. R. Š. thanks for a partial support by the ARRS Research Program P1-0297.

References

[1] M. Aouchiche, J. M. Bonnefoy, A. Fidahoussen, G. Caporossi, P. Hansen, L. Hiesse, J. Lacheré, A. Monhait, Variable neighborhood search for extremal graphs 14. The Auto–GraphiX 2 system, in: L. Liberti, N. Maculan (Eds.), *Global Optimization: From Theory to Implementation*, Springer, Berlin, 2005.

[2] A. T. Balaban, I. Motoc, D. Bonchev, O. Mekenyan, Topological indices for structure–activity correlations, *Topics Curr. Chem.* **114** (1983) 21–55.

[3] G. Caporossi, P. Hansen, Variable neighborhood search for extremal graphs. 1. The Auto–GraphiX system, *Discrete Math.* **212** (2000) 29–44.

[4] G. Caporossi, P. Hansen, Variable neighborhood search for extremal graphs. 5. Three ways to automate finding conjectures, *Discrete Math.* **276** (2004) 81–94.

[5] G. Caporossi, P. Hansen, D. Vukičević, Comparing Zagreb indices of cyclic graphs, *MATCH Commun. Math. Comput. Chem.* **63** (2010) 441–451.

[6] K. C. Das, I. Gutman, Some properties of the second Zagreb index, *MATCH Commun. Math. Comput. Chem.* **52** (2004) 103–112.

[7] K. C. Das, Sharp bounds for the sum of the squares of the degrees of a graph, *Kragujevac J. Math.* **25** (2003) 31–49.

[8] J. Devillers, A. T. Balaban, *Topological Indices and Related Descriptors in QSAR and QSPR*, Gordon and Breach, Amsterdam, 1999.

[9] I. Gutman, N. Trinajstić, Graph theory and molecular orbitals. Total $\pi-$electron energy of alternant hydrocarbons, *Chem. Phys. Lett.* **17** (1971) 535–538.

[10] I. Gutman, K. C. Das, The first Zagreb index 30 years after, *MATCH Commun. Math. Comput. Chem.* **50** (2004) 83–92.

[11] P. Hansen, D. Vukičević, Comparing the Zagreb indices, *Croat. Chem. Acta* **80** (2007) 165–168.

[12] P. Hansen, H. Mélot, I. Gutman, Variable neighborhood search for extremal graphs. 12. A note on the variance of bounded degrees in graphs, *MATCH Commun. Math. Comput. Chem.* **54** (2005) 221–232.

[13] A. Ilić, D. Stevanović, On comparing Zagreb indices, *MATCH Commun. Math. Comput. Chem.* **62** (2009) 681–687.

[14] M. Karelson, *Molecular Descriptors in QSAR/QSPR*, Wiley–Interscience, New York, 2000.

[15] B. Liu, On a conjecture about comparing Zagreb indices, in: I. Gutman, B. Furtula (Eds.), *Recent Results in the Theory of Randić Index*, Univ. Kragujevac, Kragujevac, 2008, pp. 205–209.

[16] B. Liu, I. Gutman, Upper bounds for Zagreb indices of connected graphs, *MATCH Commun. Math. Comput. Chem.* **55** (2006) 439–446.

[17] B. Liu, Z. You, A survey on comparing Zagreb indices, in: I. Gutman, B. Furtula (Eds.), *Novel Molecular Structure Descriptors – Theory and Applications I*, Univ. Kragujevac, Kragujevac, 2010, pp. 227–239.

[18] S. Nikolić, G. Kovačević, A. Miličević, N. Trinajstić, The Zagreb indices 30 years after, *Croat. Chem. Acta* **76** (2003) 113–124.

[19] L. Sun, T. Chen, Compring the Zagreb indices for graphs with small difference between the maximum and minimum degrees, *Discrete Appl. Math.* **157** (2009) 1650–1654.

[20] L. Sun, S. Wei, Comparing the Zagreb indices for connected bicyclic graphs, *MATCH Commun. Math. Comput. Chem.* **62** (2009) 699–714.

[21] R. Todeschini, V. Consonni, *Handbook of Molecular Descriptors*, Wiley–VCH, Weinheim, 2000.

[22] N. Trinajstić, *Chemical Graph Theory*, CRC Pres, Boca Raton, 1992.

[23] D. Vukičević, A. Graovac, Comparing Zagreb M_1 and M_2 indices for acyclic molecules, *MATCH Commun. Math. Comput. Chem.* **57** (2007) 587–590.

[24] D. Vukičević, A. Graovac, Comparing Zagreb M_1 and M_2 indices: Overview of the results, manuscript.

[25] D. Vukičević, Comparing Zagreb indices, talk on the meeting of International Academy of Mathematical Chemistry, Dubrovnik, 2007.

[26] S. Zhang, H. Zhang, Unicyclic graphs with the first three smallest and largest first general Zagreb index, *MATCH Commun. Math. Comput. Chem.* **55** (2006) 427–438.

[27] B. Zhou, Zagreb indices, *MATCH Commun. Math. Comput. Chem.* **52** (2004) 113–118.

[28] B. Zhou, D. Stevanović, A note on Zagreb indices, *MATCH Commun. Math. Comput. Chem.* **56** (2006) 571–577.

On Wiener index of graphs and their line graphs

Nathann Cohen [1], Darko Dimitrov [2], Roi Krakovski [3], Riste Škrekovski [4], Vida Vukašinović [5]

March 18, 2011

[1] INRIA - Projet MASCOTTE,
2004 route des Lucioles, BP 93 F-06902, Sophia Antipolis Cedex, France
nathann.cohen@sophia.inria.fr
[2] Institut für Informatik, Freie Universität Berlin,
Takustraße 9, D–14195 Berlin, Germany
darko@mi.fu-berlin.de
[3] Department of Computer Science, Ben Gurion University of the Negev,
Beer-Sheva 84105, Israel
roikr@cs.bgu.ac.il
[4] Department of Mathematics, University of Ljubljana,
Jadranska 19, 1000 Ljubljana, Slovenia
skrekovski@gmail.com
[5] Computer Systems Department, Jožef Stefan Institute,
Jamova 39, 1000 Ljubljana, Slovenia
vida.vukasinovic@ijs.si

Abstract

The Wiener index of a graph G, denoted by $W(G)$, is the sum of distances between all pairs of vertices in G. In this paper, we consider the relation between the Wiener index of a graph, G, and its line graph, $L(G)$. We show that if G is of minimum degree at least two, then $W(G) \leq W(L(G))$. We prove that for every non-negative integer g_0, there exists $g > g_0$, such that there are infinitely many graphs G of girth g, satisfying $W(G) = W(L(G))$. This partially answers a question raised by Dobrynin and Mel'nikov [8] and encourages us to conjecture that the answer to a stronger form of their question is affirmative.

Keywords: Wiener index, line graphs

1 Introduction

In this paper all graphs are finite, simple and undirected. For a graph G, we denote by $V(G)$ and $E(G)$ its vertex and edge sets, respectively. All paths and cycles are simple, i.e., they contain no repeated vertices. A path $P_n = x_1x_2 \cdots x_n$ is given by the sequence of its consecutive vertices. A path whose endvertices are u and v is called an *uv-path*. The length of a path P, denoted $|P|$, is the number of its edges. A cycle of length k is denoted by C_k.

Given a graph G, its line graph $L(G)$ is a graph such that

- The vertices of $L(G)$ are the edges of G; and
- Two vertices of $L(G)$ are adjacent if and only if their corresponding edges in G share a common endvertex.

For a vertex $v \in V(G)$, we denote by $d_G(v)$ the degree of v in G. For the sake of simplicity we write $d(v)$ if the graph G is clear from the context. For $v, u \in V(G)$, we denote by $d_G(u, v)$ (or simply $d(u, v)$), the length of a shortest path in G between u and v. For $e_1, e_2 \in E(G)$, we define $d_G(e_1, e_2) = d_{L(G)}(e_1, e_2)$.

The *Wiener index* of a graph G, denoted by $W(G)$, is the sum of distances between all (unordered) pairs of vertices of G, i.e.,

$$W(G) = \sum_{u,v \in V(G)} d(u, v).$$

The Wiener index is a graph invariant that belongs to the molecules structure-descriptors called topological indices, which are used for the design of molecules with desired properties [18]. For details and results on the Wiener index see in [6, 7, 16, 17] and the references cited therein.

The concept of line graph has various applications in physical chemistry [12, 15]. Recently there has been an interest in understanding the connection between $W(G)$ and $W(L(G))$ for a graph G. In particular, it is important to understand when a graph G satisfies $W(G) = W(L(G))$. In sequel, we state some results related to those presented in this paper. For more results on the topic see [4, 5, 9, 10, 12, 14].

Theorem 1 (Buckley [3]). *For every tree T, $W(L(T)) = W(T) - \binom{n}{2}$.*

Theorem 2 (Gutman [11]). *If G is a connected graph with n vertices and q edges, then*

$$W(L(G)) \geq W(G) - n(n-1) + \frac{1}{2}q(q+1).$$

Theorem 3 (Gutman, Pavlović [14]). *If G is a connected unicyclic graph with n vertices, then $W(L(G)) \leq W(G)$, with equality if and only if G is a cycle of length n.*

In Section 2 it will be shown that, if G is of minimum degree at least two, then, $W(G) \leq W(L(G))$, with a strict inequality as soon as G is not a cycle.

For a graph G, it seems difficult to characterize when $W(G) = W(L(G))$. Moreover, it is not clear on which graph parameters or structural properties the difference $W(G) - W(L(G))$ depends.

A connected graph G is isomorphic to $L(G)$ if and only if G is a cycle. Thus, cycles provide a trivial infinite family of graphs for which $W(G) = W(L(G))$. That is, for every positive number g there exists a graph G with girth g for which $W(G) = W(L(G))$. In connected bicyclic graphs all the three cases $W(L(G)) < W(G)$, $W(L(G)) = W(G)$, and $W(L(G)) > W(G)$ occur [14]. It is known that, the smallest bicyclic graph with the property $W(L(G)) = W(G)$ has 9 vertices and is unique. There are already 26 ten-vertex bicyclic graphs with the same property [13]. In [8], Dobrynin and Mel'nikov have constructed infinite family of graphs of girth three and four with the property $W(G) = W(L(G))$, and asked the following:

Problem 1 (Dobrynin and Mel'nikov [8]). *Is it true that for every integer $g \geq 5$, there exists a graph $G \neq C_g$ of girth g, for which $W(G) = W(L(G))$?*

The following is the main result of this paper, and provides a partial answer to Problem 1.

Theorem 4. *For every positive integer g_0, there exists $g \geq g_0$ such that there are infinitely many graphs G of girth g satisfying $W(G) = W(L(G))$.*

Our result encourages us to state the following conjecture. The answer to it for graphs of girth three and four is affirmative [8].

Conjecture 1. *For every integer $g \geq 3$, there exist infinitely many graphs G of girth g satisfying $W(G) = W(L(G))$.*

2 Graphs with minimum degree at least two

The following folk lemma is needed for the proof of Theorem 6, and states that the distance between two edges can be bounded by the mean of the distances between their endvertices. For the sake of completness we include its proof.

Lemma 5. *Let G be a graph and $e = uv$, $e' = u'v'$ be two edges of G. Then the following inequality holds:*

$$d(e, e') \geq \frac{1}{4}\Big[d(u, u') + d(u, v') + d(v, u') + d(v, v')\Big].$$

Proof. Without loss of generality, we can assume that $d(v, v') = \min\{d(u, u'), d(u, v'), d(v, u'), d(v, v')\}$. We observe that the following holds:

$$d(v, u') \leq d(v, v') + 1, \quad d(u, u') \leq d(v, v') + 2, \quad \text{and} \quad d(u, v') \leq d(v, v') + 1.$$

Therefore,

$$\frac{1}{4}\Big(d(u, u') + d(u, v') + d(v, u') + d(v, v')\Big) \leq \frac{1}{4}(4\, d(v, v') + 4) = d(v, v') + 1 = d(e, e').$$

The last equality in the above expression holds by minimality of $d(v, v')$. $\square$

The following is the main result of this section.

Theorem 6. *Let G be a connected graph with $\delta(G) \geq 2$. Then,*

$$W(G) \leq W(L(G)).$$

Moreover, equality holds only for cycles.

Proof. If G is a cycle, then $L(G)$ is isomorphic to G, and so, equality holds. Hence, we may assume that G has at least one vertex of degree at least three. By Lemma 5, we obtain a lower bound on $W(L(G))$:

$$\begin{aligned} W(L(G)) &= \sum_{\substack{e, e' \in E(G) \\ e \neq e'}} d(e, e') \\ &\geq \frac{1}{4} \sum_{\substack{e = uv \in E(G) \\ e' = u'v' \in E(G) \\ e \neq e'}} \Big(d(u, u') + d(u, v') + d(v, u') + d(v, v')\Big) \\ &= \frac{1}{4}\Bigg[\sum_{\substack{u, v \in V(G) \\ uv \notin E(G)}} d(u)d(v)d(u, v) + \sum_{\substack{u, v \in V(G) \\ uv \in E(G)}} \Big(d(u)d(v) - 1\Big) \underbrace{d(u, v)}_{=1}\Bigg]. \end{aligned}$$

Thus, for the difference $W(L(G)) - W(G)$, we obtain the following lower bound:

$$W(L(G)) - W(G) \geq \frac{1}{4}\left[\sum_{\substack{u,v \in V(G) \\ uv \notin E(G)}} d(u)d(v)d(u,v) + \sum_{\substack{u,v \in V(G) \\ uv \in E(G)}} \Big(d(u)d(v) - 1\Big)\right] - \sum_{u,v \in V(G)} d(u,v)$$
$$= \frac{1}{4}\left[\sum_{\substack{u,v \in V(G) \\ uv \notin E(G)}} \Big(d(u)d(v) - 4\Big)d(u,v) + \sum_{\substack{u,v \in V(G) \\ uv \in E(G)}} \Big(d(u)d(v) - 5\Big)\right]. \qquad (1)$$

Let G_2 be the graph induced by the vertices of degree two in G. Then,

$$\sum_{\substack{u,v \in V(G_2) \\ uv \notin E(G_2)}} \Big(d_G(u)d_G(v) - 4\Big)d_G(u,v) = 0, \qquad \text{and} \qquad \sum_{\substack{u,v \in V(G_2) \\ uv \in E(G_2)}} \Big(d_G(u)d_G(v) - 5\Big) = -|E(G_2)|. \qquad (2)$$

From (1) and (2), we obtain

$$W(L(G)) - W(G) \geq \frac{1}{4}\left[\sum_{\substack{u,v \in V(G) \\ \{u,v\} \not\subseteq V(G_2) \\ uv \notin E(G)}} \underbrace{\Big(d_G(u)d_G(v) - 4\Big)d_G(u,v)}_{\geq 1} + \sum_{\substack{u,v \in V(G) \\ \{u,v\} \not\subseteq V(G_2) \\ uv \in E(G)}} \Big(\underbrace{d_G(u)d_G(v) - 5}_{\geq 1}\Big) - |E(G_2)|\right].$$

As G has at least one vertex x of degree at least 3, the above sums are not empty. Besides, we can ensure that $|V(G_2)| - 1 \geq |E(G_2)|$: indeed, we know that $|V(H)| \geq |E(H)|$ for any graph H of maximum degree 2 with the equality holds only if H is 2-regular. But, in the present situation there is at least one vertex of degree two adjacent to a vertex of strictly larger degree in G, as the graph G is connected and G_2 is a proper subgraph of it. So, G_2 is not 2-regular, and so, $|V(G_2)| > |E(G_2)|$. Consequently,

$$W(L(G)) - W(G) \geq \frac{1}{4}\left[\sum_{v \in V(G_2)} d_G(x,v) - |V(G_2)| + 1\right] \geq \frac{1}{4}.$$

This establishes the theorem.

$\square$

3 Graphs whose Wiener index equals to the Wiener index of their line graphs

As the equality $W(L(T)) = W(T) - \binom{n}{2}$ holds for trees [3], and the equality $W(L(C)) = W(C)$ holds for cycles, one can expect that there are some graphs G, comprised of cycles and trees, with property $W(L(G)) = W(G)$. In what follows, we present one such class of graphs.

For positive integers k, p, q, we define the graph $\Phi(k, p, q)$ as follows (see Figure 1 for an illustration). The graph $\Phi(k, p, q)$ is simple and comprised of two cycles, $C_1 = u_1 \cdots u_{2k+1}$ and $C_2 = v_1 \cdots v_{2k+1}$, and two paths $P_p = x_1 \cdots x_p$ and $P_q = y_1 \cdots y_q$ such that all introduced vertices are distinct except for vertices $v_1 = u_1 = x_1$ and $y_1 = v_{2k+1} = u_{2k+1}$.

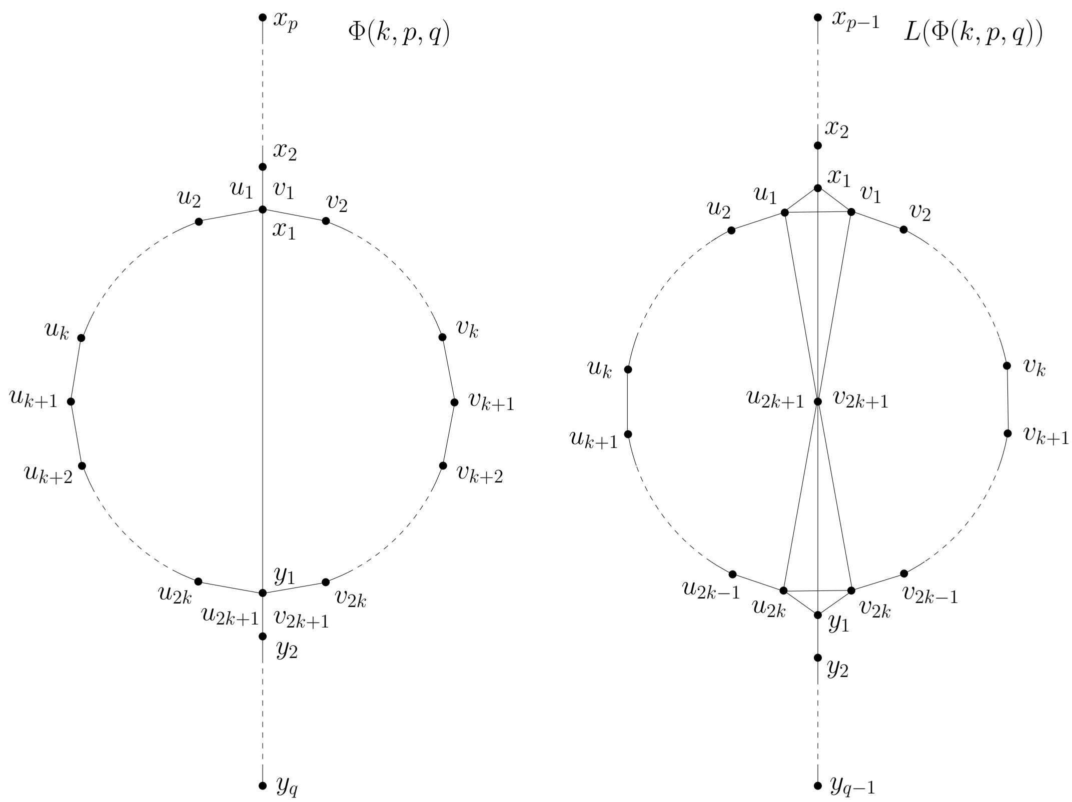


Figure 1: Graphs $\Phi(k, p, q)$ and $L(\Phi(k, p, q))$

We are now interested in computing the difference $W(L(\Phi(k, p, q))) - W(\Phi(k, p, q))$, which is determined by the following technical result, and it will be used in the proof of Theorem 8. As the proof is straightforward and rather technical, we present it in the next section.

Theorem 7. *For integers, $k, p, q \geq 1$, let $G = \Phi(k, p, q)$ with girth $g = 2k + 1$. Then,*

$$W(L(G)) - W(G) = \frac{1}{2}(g^2 + (p - q)^2 + 5(p + q - 3) - 2g(p + q - 3)).$$

We now turn to prove the main theorem of this paper.

Theorem 8. *For every non-negative integer h, there exist infinitely many graphs G of girth $g = h^2 + h + 9$ with $W(L(G)) = W(G)$.*

Proof. Our candidates are Φ graphs defined above. First we prove the following claim:

Claim 1. *Let a_0, a_1, k, such that $W(L(\Phi(k, a_0, a_1)) = W(\Phi(k, a_0, a_1))$ and $a_0 < a_1$. Then, from a_0 and a_1, we can build an infinite strictly increasing sequence $a_0, a_1, a_2, \ldots$ of integers such that for every $n \geq 0$, $W(L(\Phi(k, a_n, a_{n+1}))) = W(\Phi(k, a_n, a_{n+1}))$.*

By Theorem 7, such a sequence can only exist if the following equation is verified for all n:

$$\begin{aligned} D_n &= W(L(\Phi(k,a_n,a_{n+1}))) - W(\Phi(k,a_n,a_{n+1})) \\ &= \frac{1}{2}g^2 - ga_n - ga_{n+1} + \frac{1}{2}a_n^2 - a_na_{n+1} + \frac{1}{2}a_{n+1}^2 + 3g + \frac{5}{2}(a_n + a_{n+1}) - \frac{15}{2} = 0, \end{aligned}$$

where $g = 2k + 1$. Then,

$$\begin{aligned} D_n - D_{n+1} &= g(a_{n+2} - a_n) - \frac{1}{2}(a_{n+2}^2 - a_n^2) + a_{n+1}(a_{n+2} - a_n) - \frac{5}{2}(a_{n+2} - a_n) \\ &= (a_{n+2} - a_n)(g - \frac{1}{2}(a_{n+2} + a_n) + a_{n+1} - \frac{5}{2}) \\ &= 0. \end{aligned}$$

As we want the sequence to be strictly increasing, it is enough to solve the following recursive equation:

$$g - \frac{1}{2}(a_{n+2} + a_n) + a_{n+1} - \frac{5}{2} = 0. \tag{3}$$

It is well known that a solution to (3) is of the form $a_n = c_n + p_n$, where $c_n = nx + y$, for $x, y \in \mathbb{R}$, is the *homogeneous solution*, and $p_n = cn^2$, for $c \in \mathbb{R}$, is the *particular solution*. An easy calculation gives $y = a_0$, $x = (\frac{5}{2} + a_1 - g - a_0)$ and $c = g - \frac{5}{2}$. Hence,

$$a_n = (g - \frac{5}{2})n^2 + (\frac{5}{2} + a_1 - g - a_0)n + a_0. \tag{4}$$

Observe that for every $n \geq 0$, a_n is an integer and $a_n < a_{n+1}$. As by assumption a_0 and a_1 satisfy the equation $D_0 = 0$, the claim follows. $\Diamond$

Let k, p, q be positive integers (with $g = 2k + 1$). By Theorem 7, $W(L(\Phi(k,p,q)) = W(\Phi(k,p,q))$ if

$$g = -3 + p + q + \sqrt{24 - 11p - 11q + 4pq}. \tag{5}$$

Setting $p = 3$ and $q = h^2 + 9$ for some integer h, one obtains the equation $g = h^2 + h + 9$. Then, g is an odd positive integer. Consequently, for every $h \in \mathbb{N}$ the parameters $g = h^2+h+9$, $k = \frac{1}{2}(g - 1)$, $p = 3$, and $q = h^2 + 9$ satisfy $W(L(G)) = W(G)$. By Claim 1, for every such girth, we can compute an infinite family of graphs G satisfying the same equation by setting $a_0 = 3$ and $a_1 = h^2 + 9$. Thus, the theorem is proved.

$\square$

Clearly, the set of integer solutions of (5) is not complete (see Fig.2 for other infinite families). However, the equation (5) does not have integer solutions for every g, thus preventing us from producing an infinite family of graphs G satisfying $W(L(G)) = W(G)$ for all girths with the Φ family.

Theorem 4 is an immediate corollary of Theorem 8. For every positive integer g_0, we can choose a non-negative integer h such that $g = h^2 + h + 9 \geq g_0$. By Theorem 8, it follows that there are infinitely many graphs G of girth g with $W(L(G)) = W(G)$.

p	q	g	$24-11p-11q+4pq$
3	h^2+9	h^2+h+9	h^2
4	$20h^2+4$	$20h^2+10h+5$	$(10h)^2$
6	$13h^2+12h+6$	$13h^2+25h+15$	$(13h+6)^2$
6	$13h^2+14h+7$	$13h^2+27h+17$	$(13h+7)^2$
7	$17h^2+14h+6$	$17h^2+31h+17$	$(17h+7)^2$
7	$17h^2+20h+9$	$17h^2+37h+23$	$(17h+10)^2$
9	h^2+3	h^2+5h+9	$(5h)^2$
10	$29h^2+2h+3$	$29h^2+31h+11$	$(29h+1)^2$
10	$29h^2+56h+30$	$29h^2+85h+65$	$(29h+28)^2$
12	$37h^2+30h+9$	$37h^2+67h+31$	$(37h+15)^2$
12	$37h^2+44h+16$	$37h^2+81h+45$	$(37h+22)^2$
13	$41h^2+4h+3$	$41h^2+45h+12$	$(41h+2)^2$
13	$41h^2+78h+40$	$41h^2+119h+86$	$(41h+39)^2$
16	$53h^2+44h+12$	$53h^2+97h+41$	$(53h+22)^2$
16	$53h^2+62h+21$	$53h^2+115h+59$	$(53h+31)^2$
18	$61h^2+116h+58$	$61h^2+177h+123$	$(61h+58)^2$
18	$61h^2+128h+70$	$61h^2+189h+141$	$(61h+64)^2$

Figure 2: Families of integer solutions

4 Proof of Theorem 7

The proof of Theorem 7 follows from the following two lemmas. Their purpose is to compute the exact value of $W(G)$ and $W(L(G))$ for the Φ graphs.

Lemma 9. *Let G be a graph $\Phi(k,p,q)$ where $k,p,q \geq 1$. Then,*

$$W(G) = W(P_{p+q}) + 4W(P_{q+k}) + 4W(P_{p+k}) + 2W(C_{2k+1}) + 2W(P_{2k+1}) + 2W(P_{2k}) \\ - 16W(P_{k-1}) - 4W(P_q) - 4W(P_p) - p(p+1) - q(q+1) - 2(8k^2+k-2).$$

Proof. We consider several paths and cycles in G such that each pair of vertices of G belongs to at least one of these subgraphs. See Figure 1 for the notation. In order to make our proof more readable, we denote a shortest path between vertices a and b with $P(a,b)$. The subgraphs we consider are the following:

- The path $P(x_p, y_q) = x_p x_{p-1} \cdots x_1 y_1 y_2 \cdots y_q$ of length $p+q-1$.
- The paths $P(x_p, v_{k+1}) = x_p x_{p-1} \cdots x_2 v_1 v_2 \cdots v_{k+1}$, $P(x_p, u_{k+1}) = x_p x_{p-1} \cdots x_2 u_1 u_2 \cdots u_{k+1}$, $P(x_p, v_{k+2}) = x_p x_{p-1} \cdots x_1 v_{2k+1} v_{2k} \cdots v_{k+2}$ and $P(x_p, u_{k+2}) = x_p x_{p-1} \cdots x_1 u_{2k+1} u_{2k} \cdots u_{k+2}$ of length $p+k-1$.
- The paths $P(y_q, v_{k+1}) = y_q y_{q-1} \cdots y_2 v_{2k+1} v_{2k} \cdots v_{k+1}$, $P(y_q, u_{k+1}) = y_q y_{q-1} \cdots y_2 u_{2k+1} u_{2k} \cdots u_{k+1}$, $P(y_q, v_k) = y_q y_{q-1} \cdots y_1 v_1 v_2 \cdots v_k$ and $P(y_q, u_k) = y_q y_{q-1} \cdots y_1 u_1 u_2 \cdots u_k$ of length $q+k-1$.
- The paths $P_1(u_{k+1}, v_{k+1}) = u_{k+1} u_k \cdots u_2 v_1 v_2 \cdots v_{k+1}$ and $P_2(u_{k+1}, v_{k+1}) = u_{k+1} u_{k+2} \cdots u_{2k} v_{2k+1}\ v_{2k} \cdots v_{k+1}$ of length $2k$ have the same endvertices. Similarly, the paths $P(u_k, v_{k+2}) = u_k u_{k-1} \cdots u_1 v_{2k+1} v_{2k} \cdots v_{k+2}$ and $P(u_{k+2}, v_k) = u_{k+2}\ u_{k+3} \cdots u_{2k+1} v_1 v_2 \cdots v_k$ are of length $2k-1$.

- The cycles $C_u = u_1u_2 \cdots u_{2k+1}u_1$ and $C_v = v_1v_2 \cdots v_{2k+1}v_1$ on $2k+1$ vertices.

The following pairs of vertices were considered more than once:

- Pairs of vertices on the paths $P(x_1, x_p)$, $P(y_1, y_q)$, $P(v_2, v_k)$, $P(u_2, u_k)$, $P(v_{k+2}, v_{2k})$ and $P(u_{k+2}, u_{2k})$ are considered five times.
- Pair (x_1, y_1) is on distance 1 and is considered nine times. Similarly pair (u_{k+1}, v_{k+1}) is on distance $2k$ and is considered twice.
- Pairs (u_1, u_{k+1}), (v_1, v_{k+1}), (u_{k+1}, u_{2k+1}) and (v_{k+1}, v_{2k+1}) are on distance k. Similarly pairs of vertices $\{(u_{k+1}, a) | a \in P(u_2, u_k) \cup P(u_{k+2}, u_{2k})\}$ and $\{(v_{k+1}, a) | a \in P(v_2, v_k) \cup P(v_{k+2}, v_{2k})\}$ are on distances $1, 2, \ldots, k-1$. All of them are considered three times.
- Pairs of vertices $\{(x_1, a) | a \in P(v_2, v_k) \cup P(u_2, u_k)\}$ and $\{(y_1, a) | a \in P(v_{2k}, v_{k+2}) \cup P(u_{2k}, u_{k+2})\}$ are on distances $1, 2, \ldots, k-1$ and are considered five times.
- Pairs of vertices $\{(x_1, a) | a \in P(u_{k+2}, u_{2k}) \cup P(v_{k+2}, v_{2k})\}$ and $\{(y_1, a) | a \in P(u_2, u_k) \cup P(v_2, v_k)\}$ are on distances $2, 3, \ldots, k$ and are considered three times.
- Pairs of vertices $\{(x_1, a) | a \in P(y_2, y_q)\}$ are on distances $2, 3, \ldots, q$ and $\{(y_1, a) | a \in P(x_2, x_p)\}$ are on distances $2, 3, \ldots, p$. They are considered three times.

As the Wiener index of a graph G is the sum of the distances between all pairs of the vertices, we compute it as a sum of Wiener indices of all observed subgraphs and subtract the distances between pairs of vertices which were observed more than once. The distances are multiplied the appropriate number of times. The Wiener index of the graph $\Phi(k, p, q)$ is

$$\begin{aligned}
W(\Phi(k,p,q)) &= W(P_{p+q}) + 2W(P_{q+k}) + 2W(P_{q+k}) + 2W(P_{p+k}) + 2W(P_{p+k}) \\
&\quad + 2W(C_{2k+1}) + 2W(P_{2k+1}) + 2W(P_{2k}) - 16W(P_{k-1}) - 4W(P_q) \\
&\quad - 4W(P_p) - 8 \cdot 1 - 2k - 4 \cdot 2 \cdot k - 4 \cdot 2(1 + 2 + \cdots + k - 1) \\
&\quad - 4 \cdot 4(1 + 2 + \cdots + k - 1) - 4 \cdot 2(2 + 3 + \cdots + k) - 2(2 + 3 + \cdots + q) \\
&\quad - 2(2 + 3 + \cdots + p) \\
&= W(P_{p+q}) + 4W(P_{q+k}) + 4W(P_{p+k}) + 2W(C_{2k+1}) + 2W(P_{2k+1}) \\
&\quad + 2W(P_{2k}) - 16W(P_{k-1}) - 4W(P_q) - 4W(P_p) - p^2 - p - q^2 - q \\
&\quad - 16k^2 - 2k + 4.
\end{aligned}$$

□

Lemma 10. *Let $G = \Phi(k, p, q)$ where $k, p, q \geq 1$. Then,*

$$\begin{aligned}
W(L(G)) &= W(P_{p+q-1}) + 2W(P_{q+k-1}) + 2W(P_{q+k}) + 2W(P_{p+k-1}) + 2W(P_{p+k}) \\
&\quad + 2W(C_{2k+1}) + 2W(P_{2k}) + 2W(P_{2k+1}) - 16W(P_k) - 4W(P_{q-1}) \\
&\quad - 4W(P_{p-1}) - p(p-1) - q(q-1) - 4k(k+1).
\end{aligned}$$

Proof. Similar as in the previous lemma, we consider paths and cycles in $L(\Phi(k, p, q))$ such that each pair of vertices $L(\phi(k, p, q))$ belongs to at least one of these subgraphs. The subgraphs we consider are the following:

- The path $P(x_{p-1}, y_{q-1}) = x_{p-1}x_{p-2}\cdots x_1v_{2k+1}y_1y_2\cdots y_{q-1}$ of length $p+q-2$.
- The paths $P(x_{p-1}, v_k) = x_{p-1}x_{p-2}\cdots x_1v_1v_2\cdots v_k$, $P(x_{p-1}, u_k) = x_{p-1}x_{p-2}\cdots x_1u_1 u_2\cdots u_k$ of length $p+k-2$ and the paths $P(x_{p-1}, v_{k+1}) = x_{p-1}x_{p-2}\cdots x_1v_{2k+1}v_{2k}\cdots v_{k+1}$, $P(x_{p-1}, u_{k+1}) = x_{p-1}\ x_{p-2}\cdots x_1u_{2k+1}u_{2k}\cdots u_{k+1}$ of length $p+k-1$.
- The paths $P(y_{q-1}, v_{k+1}) = y_{q-1}y_{q-2}\cdots y_1v_{2k}v_{2k-1}\cdots v_{k+1}$, $P(y_{q-1}, u_{k+1}) = y_{q-1}y_{q-2}\cdots y_1u_{2k}u_{2k-1}\cdots u_{k+1}$ of length $q+k-2$ and the paths $P(y_{q-1}, v_k) = y_{q-1}y_{q-2}\cdots y_1v_{2k+1} v_1v_2\cdots v_k$, $P(y_{q-1}, u_k) = y_{q-1}y_{q-2}\cdots y_1\ u_{2k+1}u_1u_2\cdots u_k$ of length $q+k-1$.
- The paths $P(u_k, v_k) = u_ku_{k-1}\cdots u_1v_1v_2\cdots v_k$, $P(u_{k+1}, v_{k+1}) = u_{k+1}u_{k+2}\cdots u_{2k}v_{2k} v_{2k-1}\cdots v_{k+1}$ of length $2k-1$ and the paths $P(u_k, v_{k+1}) = u_ku_{k-1}\cdots u_1v_{2k+1}v_{2k}\cdots v_{k+1}$, $P(u_{k+1}, v_k) = u_{k+1}u_{k+2}\ \cdots u_{2k+1}v_1v_2\cdots v_k$ of length $2k$.
- The cycles $C_u = u_1u_2\cdots u_{2k+1}u_1$ and $C_v = v_1v_2\cdots v_{2k+1}v_1$ on $2k+1$ vertices.

The pairs of vertices which were observed more than once are the following:

- Pairs of vertices on the paths $P(x_1, x_{p-1})$, $P(y_1, y_{q-1})$, $P(v_1, v_k)$, $P(u_1, u_k)$, $P(v_{k+1}, v_{2k})$ and $P(u_{k+1}, u_{2k})$ are considered five times.
- Pairs of vertices $\{(v_{2k+1}, a) | a \in P(u_1, u_k) \cup P(v_1, v_k) \cup P(u_{k+1}, u_{2k}) \cup P(v_{k+1}, v_{2k})\}$ are on distances $1, 2, \ldots, k$ and are considered three times.
- Pairs of vertices $\{(v_{2k+1}, a) | a \in P(y_1, y_{q-1})\}$ are on distances $1, 2, \ldots, q-1$ and are considered three times. Similarly pairs of vertices $\{(v_{2k+1}, a) | a \in P(x_1, x_{p-1})\}$ are on distances $1, 2, \ldots, p-1$ and are considered three times.

The Wiener index is calculated as a difference between a sum of Wiener indices of all observed subgraphs and corresponding multiplication of distances between different pairs of vertices which were observed more than once:

$$\begin{aligned}
W(L(G)) &= W(P_{p+q-1}) + 2W(P_{q+k-1}) + 2W(P_{q+k}) + 2W(P_{p+k-1}) + 2W(P_{p+k}) \\
&\quad + 2W(C_{2k+1}) + 2W(P_{2k}) + 2W(P_{2k+1}) - 16W(P_k) - 4W(P_{q-1}) \\
&\quad - 4W(P_{p-1}) - 4 \cdot 2(1 + 2 + \cdots + k) \\
&\quad - 2(1 + 2 + \cdots + q - 1) - 2(1 + 2 + \cdots + p - 1)) \\
&= W(P_{p+q-1}) + 2W(P_{q+k-1}) + 2W(P_{q+k}) + 2W(P_{p+k-1}) + 2W(P_{p+k}) \\
&\quad + 2W(C_{2k+1}) + 2W(P_{2k}) + 2W(P_{2k+1}) - 16W(P_k) - 4W(P_{q-1}) \\
&\quad - 4W(P_{p-1}) - p^2 + p - q^2 + q - 4k^2 - 4k.
\end{aligned}$$

□

Proof of Theorem 7. By Lemmas 9 and 10, it follows that

$$\begin{aligned}
W(L(G)) - W(G) &= W(P_{p+q-1}) - W(P_{p+q}) + 2(W(P_{q+k-1}) - W(P_{q+k})) + 2(W(P_{p+k-1}) \\
&\quad - W(P_{p+k})) + 4(W(P_q) - W(P_{q-1})) + 4(W(P_p) - W(P_{p-1})) \\
&\quad + 16(W(P_{k-1}) - W(P_k)) + p + q - 4k^2 - 4k + p + q + 16k^2 + 2k - 4.
\end{aligned}$$

The Wiener index of a path with n vertices being $W(P_n) = \binom{n+1}{3}$ [2], we have

$$\begin{aligned}
W(L(G)) - W(G) &= \binom{p+q}{3} - \binom{p+q+1}{3} + 2\left(\binom{q+k}{3} - \binom{q+k+1}{3}\right) \\
&\quad + 2\left(\binom{p+k}{3} - \binom{p+k+1}{3}\right) + 4\left(\binom{q+1}{3} - \binom{q}{3}\right) \\
&\quad + 4\left(\binom{p+1}{3} - \binom{p}{3}\right) + 16\left(\binom{k}{3} - \binom{k+1}{3}\right) \\
&\quad + 2(p+q) + 12k^2 - 2k - 4 \\
&= -\binom{p+q}{2} - 2\binom{q+k}{2} - 2\binom{p+k}{2} + 4\binom{q}{2} + 4\binom{p}{2} - 16\binom{k}{2} \\
&\quad + 2(p+q) + 12k^2 - 2k - 4 \\
&= \frac{1}{2}(-8 + 4k^2 + 3p + (p-q)^2 + 3q - 4k(-4+p+q)).
\end{aligned}$$

If we set $k = (g-1)/2$, we obtain the claimed formula

$$W(L(G)) - W(G) = \frac{1}{2}(g^2 + (p-q)^2 + 5(p+q-3) - 2g(p+q-3)).$$

□

4.1 Wiener index and Combinatorial Nullstellensatz

We bring to reader's attention the fact that the polynomials given in Theorem 8 can be easily obtained through polynomial interpolation with the help of a computer. Indeed, the above proofs can be massively shortened and simplified if one only needs to show that both $W(G)$ and $W(L(G))$ are low-degree polynomials on the variables k, p and q.

Once bounds on the degree of each variable in the polynomials $W(L(\Phi(k,p,q)))$ and $W(\Phi(k,p,q))$ have been derived, it is easy to define a (small) set of representatives of the Φ family which are sufficient to define exactly the corresponding polynomials using the Combinatorial Nullstellensatz [1] (less than 30 different graphs in the present case).

This way, a computer can be made to answer very quickly the following question: "given a graph family $\mathcal{G}$ depending on several parameters $p_1, \ldots, p_l$, what is the general formula of $W(\mathcal{G}(p_1, \ldots, p_l)) - W(L(\mathcal{G}(p_1, \ldots, p_l)))$?". This is of great help when looking for graphs G satisfying the equation $W(G) = W(L(G))$, as it reduces the problem to finding the integral zeros of a multivariate polynomial (which is not by itself an easy question).

This approach has to be considered when trying to find more classes of graphs satisfying the above constraint, especially when the Φ family used here can be modified in so many ways: one could like to attach paths to the cycles at different points, set two different sizes for the cycles, or to attach trees instead of paths, etc.

References

[1] N. Alon. Combinatorial Nullstellensatz. *Combin. Probab. Comput.*, 8(1&2):7–29, 1999.

[2] D. Babić, D. J. Klein, I. Lukovits, S. Nikolić, and N. Trinajstić. Resistance-distance matrix: A computational algorithm and its applications. *Int. J. Quant. Chem.*, 90:166–176, 2002.

[3] F. Buckley. Mean distance of line graphs. *Congr. Numer.*, 32:153–162, 1981.

[4] P. Dankelmann, I. Gutman, S. Mukwembi, and H. C. Swart. The edge-Wiener index of a graph. *Discrete Math.*, 309:3452–3457, 2009.

[5] A. A. Dobrynin. Distance of iterated line graphs. *Graph Theory Notes N. Y.*, 37:8–9, 1999.

[6] A. A. Dobrynin, R. Entringer, and I. Gutman. Wiener index of trees: theory and applications. *Acta Appl. Math.*, 66:211–249, 2001.

[7] A. A. Dobrynin, I. Gutman, S. Klavžar, and P. Žigert. Wiener index of hexagonal systems. *Acta Appl. Math.*, 72:247–294, 2002.

[8] A. A. Dobrynin and L. S. Melnikov. Some results on the Wiener index of iterated line graphs. *Electron. Notes Discrete Math.*, 22:469–475, 2005.

[9] A. A. Dobrynin and L. S. Melnikov. Wiener index for graphs and their line graphs with arbitrary large cyclomatic numbers. *Appl. Math. Lett.*, 18(3):307–312, 2005.

[10] A. A. Dobrynin and L. S. Melnikov. Wiener index, line graph and the cyclomatic number. *MATCH Commun. Math. Comput. Chem.*, 53:209–214, 2005.

[11] I. Gutman. Distance of line graphs. *Graph Theory Notes N. Y.*, 31:49–52, 1996.

[12] I. Gutman and E. Estrada. Topological indices based on the line graph of the molecular graph. *J. Chem. Inf. and Comput. Sci.*, 36(3):541–543, 1996.

[13] I. Gutman, V. Jovašević, and A. Dobrynin. Smallest graphs for which the distance of the graph is equal to the distance of its line graph. *Graph Theory Notes N. Y.*, 33:19, 1997.

[14] I. Gutman and L. Pavlović. More on distance of line graphs. *Graph Theory Notes N. Y.*, 33:14–18, 1997.

[15] I. Gutman, L. Popović, B. K. Mishra, M. Kuanar, E. Estrada, and N. Guevara. Application of line graphs in physical chemistry. Predicting the surface tensions of alkanes. *J. Serb. Chem. Soc.*, 62(3):1025–1029, 1997.

[16] I. Gutman, Y. Yeh, S. Lee, and Y. Luo. Some recent results in the theory of the Wiener number. *Indian J. Chem.*, 32A:651–661, 1993.

[17] S. Nikolić, N. Trinajstić, and Z. Mihalić. The Wiener index: developments and applications. *Croat. Chem. Acta*, 68:105–129, 1995.

[18] M. Randić. In search for graph invariants of chemical interest. *J. Mol. Struct.*, 300:551–571, 1993.

DMFA Logo AMC Logo

Also available at http://amc.imfm.si

ARS MATHEMATICA CONTEMPORANEA x (xxxx) 1–x

A note on Zagreb indices inequality for trees and unicyclic graphs*

Vesna Andova

Faculty of Electrical Engineering and Information Technologies, Ss Cyril and Methodius Univ., Ruger Boskovik PO Box 574, 1000 Skopje, Macedonia

Nathann Cohen †

Projet Mascotte, I3S(CNRS, UNSA) and INRIA, 2004 route des lucioles, BP 93, 06902 Sophia-Antipolis Cedex, France

Riste Škrekovski ‡

Department of Mathematics, University of Ljubljana, Jadranska 21, 1111 Ljubljana, Slovenia

Received dd mmmm yyyy, accepted dd mmmmm yyyy, published online dd mmmmm yyyy

Abstract

For a simple graph G with n vertices and m edges, the inequality $\frac{M_1(G)}{n} \leq \frac{M_2(G)}{m}$, where $M_1(G)$ and $M_2(G)$ are the first and the second Zagreb indices of G, is known as Zagreb indices inequality. Recently Vukičević and Graovac [12], and Caporossi, Hansen and Vukičević [3] proved that this inequality holds for trees and unicyclic graphs, respectively. Here, alternative and shorter proofs for trees and for unicyclic graphs are presented.

Keywords: *First Zagreb index, second Zagreb index*

Math. Subj. Class.: 05C05, 05C07,05C38, 92E10, 94C15

1 Introduction

The first and second Zagreb indices are among the oldest topological indices, defined in 1972 by Gutman et al. [5], and are given different names in the literature, such as the Zagreb group indices, the Zagreb group parameters and most often, the Zagreb indices.

*Partial support by bilateral project BI-FR/09-10-PROTEUS-008.

†Partially supported by the ANR Blanc AGAPE and ANR Blanc International-Taiwan GRATEL.

‡Supported in part by ARRS Research Program P1-0297.

E-mail addresses: vesna.andova@gmail.com (Vesna Andova), nathann.cohen@gmail.com (Nathann Cohen), skrekovski@gmail.com (Riste Škrekovski)

Since then, they have been used to study molecular complexity, chirality, ZE-isomerism and hetero-systems (see [1, 4, 7, 9, 14]).

In the following, let $G = (V, E)$ be a simple graph with $n = |V|$ vertices and $m = |E|$ edges. These indices are defined as

$$M_1(G) = \sum_{v \in V} d(v)^2 \quad \text{and} \quad M_2(G) = \sum_{uv \in E} d(u)d(v),$$

where $d(u)$ stands for the degree of vertex u. For the sake of simplicity, we will often use M_1 and M_2 instead of $M_1(G)$ and $M_2(G)$, respectively.

In 2003, an article [8] repopularized Zagreb indices, and since then a lot of work was done on this topic. For more results concerning Zagreb indices see [10, 13]. Comparing the values of these indices on the same graph was one very natural aim, which gave, and still gives, very interesting results. At first the next conjecture was proposed [2]:

Conjecture 1.1. *For all simple graphs G,*

$$\frac{M_1(G)}{n} \le \frac{M_2(G)}{m} \tag{1.1}$$

and the bound is tight for complete graphs.

If the graph is regular then this bound is tight, but it is also tight if G is a star. This inequality holds for trees [12], graphs of maximum degree four, i.e. so called chemical graphs [6] and unicyclic graphs [3], but does not hold in general. See [6, 12, 3, 11] for various examples of graphs dissatisfying the inequality (1.1).

For a connected graph G, the *cyclomatic* number is $\nu(G) = m - n + 1$. Thus, every tree has cyclomatic number 0. A graph whose cyclomatic number is 1 is called *unicyclic*. Note that such a graph has precisely one cycle.

In chemistry trees, unicyclic graphs, bicyclic graphs, and so on, are very important graphs since they represent classes of molecules. Trees are graph representation of acyclic molecules like alkanes (also known as paraffins). Cycloalkanes are types of alkanes which have one or more rings of carbon atoms in the chemical structure of their molecules, so their graphs are unicyclic graphs, bicyclic graphs, etc.

In this paper we present alternative proofs concerning the Zagreb indices inequality for trees and unicyclic graphs.

2 An alternative proof for trees and unicyclic graphs

As we said before, trees and unicyclic graphs satisfy $M_1/n \le M_2/m$. Here, these results are proven in a shorter way.

A star with k edges is called a *k-star*. A path of length k is called a *k-path*. Let $p_3(G)$ be the number of 3-paths, $p_2(G)$ the number of 2-paths, and $C_3(G)$ is the number of 3-cycles in G. Note that

$$p_3(G) + 3C_3(G) = \sum_{uv \in E} (d(v) - 1)(d(u) - 1), \tag{2.1}$$

where uv in the summation is the middle edge of the $(d(u) - 1)\,(d(v) - 1)$ corresponding 3-paths. Obviously, a 3-path corresponds to a 3-cycle when its endvertices coincide.

Theorem 2.1. *For any tree $G \neq K_1$, it holds $\frac{M_1}{n} \leq \frac{M_2}{m}$. Moreover, equality holds if and only if G is a star.*

Proof. If G is a k-star, then $M_1 = kn$ and $M_2 = km$, by which we have equality in (1.1). So assume now that G has at least two internal adjacent vertices u and v and that v is the only internal neighbor of u. Observe that $M_1 = \sum_{v \in V} d(v)^2 = 2(p_2(G) + m)$. We have

$$M_2 = \sum_{uv \in E} \Big[(d(v)-1)(d(u)-1) + (d(u)+d(v)) - 1\Big] = p_3(G) + M_1 - m. \tag{2.2}$$

Now, since $m = n-1$, we obtain

$$\begin{aligned} (n-1)M_1 &< nM_2 \\ (n-1)M_1 &< n\left[p_3(G) + M_1 - (n-1)\right] \\ 0 &< p_3(G) + \frac{2}{n}\left(p_2(G) + (n-1)\right) - (n-1). \end{aligned}$$

Notice that $p_2(G) \geq 2$ for every tree on at least 4 vertices. Now, we will prove that $p_3(G) \geq n-3$, and this will establish the theorem. Let $l_1, \ldots, l_k$ be the leaves adjacent to u, and let $w \neq u$ be a neighbor of v. To any vertex x at distance at least 2 from u we associate the 3-path built from the first three edges of the shortest path from x to l_1. To any leaf l_i, $(i \neq 1)$, we associate the path from w to l_i. These 3-paths being all different, we associated a 3-path to any vertex except three, namely l_1, u, v, which ensures that $p_3(G) \geq n-3$. □

Theorem 2.2. *For any unicyclic graph G, it holds $\frac{M_1}{n} \leq \frac{M_2}{m}$. Moreover, equality holds if and only if G is a cycle.*

Proof. Since G is an unicyclic graph, $m = n$, and so we need to show $M_1 \leq M_2$. If G is a k-cycle then $M_1 = 4k$ and $M_2 = 4k$, so we have equality in (1.1). So, assume that G is not a cycle, $C = x_1 x_2 \cdots x_l x_1$ is the unique cycle of G and x_1, has a neighbor $y \notin V(C)$. From (2.1) and the left equality of (2.2), we have

$$M_2 = p_3(G) + 3C_3(G) + M_1 - m.$$

It is enough to show that $M_1 + 1 \leq M_2$ which is equivalent to $M_1 \leq p_3(G) + 3C_3(G) + M_1 - n - 1$, and hence is equivalent to

$$n + 1 \leq p_3(G) + 3C_3(G). \tag{2.3}$$

Now, remove the edge x_1x_2 from the cycle. Then $G - x_1x_2$ is a tree and $p_3(G - x_1x_2) \geq n-3$. Including $yx_1x_2x_3$ we have at least $n-2$ different 3-paths.

If C is a 3-cycle, then it is obvious that (2.3) holds. Now, assume $l \geq 4$. Observe that $x_1x_2x_3x_4$, $x_lx_1x_2x_3$, $x_{l-1}x_lx_1x_2$ are 3-paths all distinct from the 3-paths described. Hence, $p_3(G) \geq n+1$. This implies (2.3). □

References

[1] D. Bonchev, N. Trinajstić, Overall Molecular Descriptors. 3. Overall Zagreb Indices, *SAR&QSAR Environ. Res.* **12**(1) (2001) 213–236.

[2] G. Caporossi, P. Hansen, Variable Neighborhood search for extremal graphs. 1. The AutoGraphiX system, *Discrete Math.* **212** (2000) 29–44.

[3] G. Caporossi, P. Hansen, D. Vukičević, Comparing Zagreb indices of cyclic graphs, *MATCH Commun. Math. Comput. Chem.* **63** (2010) 441–451.

[4] I. Gutman, B. Rušćić, N. Trinajstić, C. F. Wilcox, Jr., Graph Theory and Molecular Orbitals. XII. Acyclic Polyenes, *J. Chem. Phys.* **62** (1975) 3399–3405.

[5] I. Gutman, N. Trinajstić, Graph theory and molecular orbitals. Total $\pi-$electron energy of alternant hydrocarbons, *Chem. Phys. Lett.* **17** (1972) 535–538.

[6] P. Hansen, D. Vukičević, Comparing the Zagreb Indices, *Croat. Chem. Acta* **80** (2007) 165–168.

[7] A. Miličević, S. Nikolić, N. Trinajstić, On Reformulated Zagreb Indices, *Molecular Diversity* **8** (2004) 393–399.

[8] S. Nikolić, G. Kovačević, A. Miličević, N. Trinajstić, The Zagreb Indices 30 years after, *Croat. Chem. Acta* **76** (2003) 113–124.

[9] S. Nikolić, I. M. Tolić, N. Trinajstić, I. Baučic, On the Zagreb Indices as Complexity Indices, *Croat. Chem. Acta* **73** (2000) 909–921.

[10] B. Liu, Z. You, A survey on comparing Zagreb indices, in: I. Gutman, B. Furtula (eds), *Novel Molecular Structure Descriptors – Theory and Applications I*, Univ. Kragujevac, Kragujevac, 2010, 227–239.

[11] L. Sun, S. Wei, Comparing the Zagreb indices for connected bicyclic graphs, *MATCH Commun. Math. Comput. Chem.* **62** (2009) 699–714.

[12] D. Vukičević, A. Graovac, Comparing Zagreb M_1 and M_2 indices for acyclic molecules, *MATCH Commun. Math. Comput. Chem.* **57**(3) (2007) 587–590.

[13] D. Vukičević, A. Graovac, Comparing Zagreb M_1 and M_2 indices: Overview of the results, manuscript.

[14] D. Vukičević, S. Nikolić, N. Trinajstić, On the Path-Zagreb Matrix, *J. Math. Chem.* **45** (2009) 538–543.

Nomenclature

α	Maximum size of an independent set
χ	Chromatic number (see p.14)
χ'	Chromatic index (see p.25)
χ'_a	Acyclic chromatic index
χ_a	Acyclic chromatic number
χ_f	Fractional chromatic number (see p.19)
Δ	Maximum degree of a vertex
δ^*	Degeneracy (see p.17)
Δ^+	Maximum outdegree of a vertex
$\mathcal{G}_p^n$	Random graph on n vertices and probability p
ω	Clique number (maximum size of an complete subgraph)
ch	Vertex choosability (see p.20)
ch'	Edge choosability (see p.29)
g	Girth (see p.18, 35)
K_k^n	Kneser's graph defined over $\binom{n}{k}$ (see p.13)
la	Linear Arboricity
mad	Maximum average degree (see p.32)

Bibliography

[1] B.D. Acharya and M. Las Vergnas. Hypergraphs with cyclomatic number zero, triangulated graphs, and an inequality. *Journal of Combinatorial Theory, Series B*, 33(1):52–56, 1982.

[2] J. Akiyama, G. Exoo, and F. Harary. Covering and packing in graphs. III: Cyclic and acyclic invariants. *Mathematica Slovaca*, 30(4):405–417, 1980.

[3] M.O. Albertson and D.M. Berman. The acyclic chromatic number. In *Proc. Seventh Southeastern Conference on Combinatorics, Graph Theory and Computing, Utilitas Mathematica Inc., Winniepeg, Canada*, pages 51–60, 1976.

[4] N. Alon. Non-constructive proofs in combinatorics. In *Proc. of the International Congress of Mathematicians, Kyoto*, pages 1421–1429. Citeseer, 1990.

[5] N. Alon. Combinatorial nullstellensatz. *Combinatorics, Probability and Computing*, 8(1&2):7–29, 1999.

[6] N. Alon, J. Bruck, J. Naor, M. Naor, and R.M. Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *Information Theory, IEEE Transactions on*, 38(2):509–516, 1992.

[7] N. Alon, C. McDiarmid, and B. Reed. Acyclic coloring of graphs. *Random Structures and Algorithms*, 2(3):277–288, 1991.

[8] N. Alon and M. Naor. Derandomization, witnesses for Boolean matrix multiplication and construction of perfect hash functions. *Algorithmica*, 16(4):434–449, 1996.

[9] N. Alon and J.H. Spencer. *The probabilistic method.* Wiley-Interscience, 2000.

[10] N. Alon, B. Sudakov, and A. Zaks. Acyclic edge colorings of graphs. *Journal of Graph Theory*, 37(3):157–167, 2001.

[11] N. Alon and M. Tarsi. Colorings and orientations of graphs. *Combinatorica*, 12(2):125–134, 1992.

[12] N. Alon, VJ Teague, and NC Wormald. Linear arboricity and linear k-arboricity of regular graphs. *Graphs and Combinatorics*, 17(1):11–16, 2001.

[13] N. Alon and R. Yuster. On a hypergraph matching problem. *Graphs and Combinatorics*, 21(4):377–384, 2005.

[14] N. Alon, R. Yuster, and U. Zwick. Color-coding. *Journal of the ACM (JACM)*, 42(4):856, 1995.

[15] Noga Alon, Colin Mcdiarmid, and Bruce Reed. Acyclic coloring of graphs. *Random Structures & Algorithms*, 2(3):277–288, 1991.

[16] V. Andova, N. Cohen, and R. Skrekovski. A note on zagreb indices inequality for trees and unicyclic graphs. *(manuscript)*, 2010.

[17] V. Andova, N. Cohen, and R. Skrekovski. Graph classes (dis)satisfying the zagreb indices inequality. *MATCH Commun. Math. Comput. Chem.*, 65(3):647–658, 2011.

[18] K.I. Appel and W. Haken. *The solution of the four-color-map problem*. Freeman, 1977.

[19] J. Araujo, N. Cohen, F. Giroire, and F. Havet. Good edge-labelling of graphs. In *proceedings of the Latin-American Algorithms, Graphs and Optimization Symposium (LAGOS'09)*, volume 35 of *Electronic Notes in Discrete Mathematics*, pages 275–280, Gramado, Brazil, December 2009. Springer.

[20] S. Arnborg and A. Proskurowski. Linear time algorithms for np-hard problems restricted to partial k-trees. *Discrete Applied Mathematics*, 23(1):11–24, 1989.

[21] M Basavaraju and L. Sunil Chandran. Acyclic edge coloring of 2-degenerate graphs. *Journal of Graph Theory*, 2010.

[22] Manu Basavaraju, L. Sunil Chandran, Nathann Cohen, Frederic Havet, and Tobias Müller. Acyclic edge-colouring of planar graphs. *To appear in SIAM Journal on Discrete Mathematics*.

[23] C. Beeri, R. Fagin, D. Maier, and M. Yannakakis. On the desirability of acyclic database schemes. *Journal of the ACM (JACM)*, 30(3):479–513, 1983.

[24] L.W. Beineke. Characterizations of derived graphs. *Journal of Combinatorial Theory*, 9(2):129–135, 1970.

[25] C. Berge. *Graphs*. North-Holland, 1985.

[26] C. Berge. *Hypergraphs: combinatorics of finite sets*. North-Holland, 1989.

[27] J.-C. Bermond, Y. M. Chee, N. Cohen, and X. Zhang. The α-arboricity of complete uniform hypergraphs. *SIAM Journal on Discrete Mathematics*, 25(2):600–610, 2011.

[28] J.-C. Bermond, M. Cosnard, and S. Pérennes. Directed acyclic graphs with unique path property. Technical Report RR-6932, INRIA, May 2009.

[29] H. Bodlaender. Treewidth: Algorithmic techniques and results. *Mathematical Foundations of Computer Science 1997*, pages 19–36, 1997.

[30] B. Bollobás. *Combinatorics: set systems, hypergraphs, families of vectors, and combinatorial probability*. Cambridge University Press, 1986.

[31] B. Bollobás. *Random graphs*. Cambridge university press, 2001.

[32] B. Bollobas and AJ Harris. List-colourings of graphs. *Graphs and Combinatorics*, 1(1):115–127, 1985.

[33] A. Bondy and J. Szwarcfiter. Induced decompositions of graphs. *submitted*, 2010.

[34] J.A. Bondy and U.S.R. Murty. *Graph theory with applications*, volume 290. MacMillan London, 1976.

[35] OV Borodin. Criterion of chromaticity of a degree prescription. In *IV All-Union Conf. on Theoretical Cybernetics (Novosibirsk)*, pages 127–128, 1977.

[36] OV Borodin. A generalization of Kotzig's theorem and prescribed edge coloring of planar graphs. *Mat. Zametki*, 48:22–28, 1990.

[37] O.V. Borodin, A.V. Kostochka, and D.R. Woodall. List edge and list total colourings of multigraphs. *Journal of combinatorial theory, Series B*, 71(2):184–204, 1997.

[38] J.M. Boyer and W.J. Myrvold. On the cutting edge: Simplified o (n) planarity by edge addition. *Journal of Graph Algorithms and Applications*, 8(3):241–273, 2004.

[39] A. Brandstädt, VB Le, T. Szymczak, F. Siegemund, HN de Ridder, S. Knorr, M. Rzehak, M. Mowitz, and N. Ryabova. ISGCI: Information System on Graph Class Inclusions.

[40] RL Brooks. On colouring the nodes of a network. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 37, pages 194–197. Cambridge Univ Press, 1941.

[41] M. Burlet, F. Maffray, and N. Trotignon. Odd pairs of cliques. *Graph Theory in Paris*, pages 85–95, 2007.

[42] L. Caccetta and R. Haggkvist. *On minimal digraphs with given girth*. Dept. of Combinatorics and Optimization, University of Waterloo, 1978.

[43] P.J. Cameron. Encyclopaedia of designtheory. 2006. *Available from World Wide Web: http://designtheory. org/library/encyc*.

[44] C. Capelle, M. Habib, and F. De Montgolfier. Graph decompositions and factorizing permutations. *Discrete Mathematics and Theoretical Computer Science*, 5(1):55–70, 2002.

[45] G. Caporossi, P. Hansen, and D. Vukičević. Comparing zagreb indices of cyclic graphs. *MATCH Commun. Math. Comput. Chem.*, 63:441–451, 2010.

[46] Alexandre Casamayou, Nathann Cohen, Guillaume Connan, Thierry Dumont, Laurent Fousse, Franccois Maltey, Matthias Meulien, Marc Mezzarobba, Clément Pernet, Nicolas M. Thiéry, and Paul Zimmermann. *Calcul mathématique avec Sage*. 2010.

[47] Coin-or branch-and-cut mip solver. `https://projects.coin-or.org/Cbc`.

[48] V. Chvátal. Intersecting families of edges in hypergraphs having the hereditary property. In *Hypergraph Seminar*, pages 61–66. Springer, 1974.

[49] N. Cohen, D. Dimitrov, R. Krakovski, R. Skrekovski, and V. Vukasinovic. On wiener index of graphs and their line graphs. *MATCH Commun. Math. Comput. Chem.*, 64(3):683–698, 2010.

[50] N. Cohen, F. V. Fomin, G. Gutin, E. Jung Kim, S. Saurabh, and A. Yeo. Algorithm for finding k-vertex out-trees and its application to k-internal out-branching problem. *Journal of Computer and System Sciences*, 76(7):650 – 662, 2010.

[51] N. Cohen and F. Havet. Planar graphs with maximum degree $\delta \geq 9$ are $(\delta+1)$-edge-choosable – short proof. *Discrete Mathematics*, 310(21):3049–3051, 2010.

[52] Nathann Cohen, David Coudert, Dorian Mazauric, Napoleão Nepomuceno, and Nicolas Nisse. Tradeoffs in process strategy games with application in the wdm reconfiguration problem. *Theoretical Computer Science*, 412(35):4675 – 4687, 2011.

[53] Nathann Cohen and Frédéric Havet. Linear and 2-frugal choosability of graphs of small maximum average degree. *Graphs and Combinatorics*, 27:831–849, 2011. 10.1007/s00373-010-1009-7.

[54] C.J. Colbourn and J.H. Dinitz. *Handbook of combinatorial designs*. CRC press, 2006.

[55] R. Cole, Ł. Kowalik, and R. Škrekovski. A generalization of Kotzig's theorem and its application. *SIAM Journal on Discrete Mathematics*, 21:93, 2007.

[56] D. Coudert, S. Perennes, Q.C. Pham, and J.S. Sereni. Rerouting requests in wdm networks. *AlgoTel 2005 Septiemes rencontres francopohones sur les Aspects Algorithmiques des Telecommunications*, page 17, 2005.

[57] Ibm ilog cplex optimizer. `www.ibm.com/software/integration/optimization/cplex-optimizer/`.

[58] Marek Cygan, Jian-Feng Hou, Łukasz Kowalik, Borut Lužar, and Jian-Liang Wu. A planar linear arboricity conjecture. *Journal of Graph Theory*, 2011.

[59] D.P. Dailey. Uniqueness of colorability and colorability of planar 4-regular graphs are np-complete* 1. *Discrete Mathematics*, 30(3):289–293, 1980.

[60] P. Dankelmann, I. Gutman, S. Mukwembi, and H. C. Swart. The edge-Wiener index of a graph. *Discrete Math.*, 309:3452–3457, 2009.

[61] Fabien de Montgolfier. Modular decomposition code. `http://www.liafa.jussieu.fr/ fm/algos/index.html`.

[62] R. Diestel. Graph theory, volume 173 of Graduate Texts in Mathematics. *Springer, Heidelberg*, 91:92, 2005.

[63] A. A. Dobrynin. Distance of iterated line graphs. *Graph Theory Notes N. Y.*, 37:8–9, 1999.

[64] Andrey A. Dobrynin and Leonid S. Mel'nikov. Some results on the Wiener index of iterated line graphs. *Electron. Notes Discrete Math.*, 22:469–475, 2005.

[65] Andrey A. Dobrynin and Leonid S. Melnikov. Wiener index for graphs and their line graphs with arbitrary large cyclomatic numbers. *Appl. Math. Lett.*, 18(3):307–312, 2005.

[66] Andrey A. Dobrynin and Leonid S. Mel'nikov. Wiener index, line graph and the cyclomatic number. *MATCH Commun. Math. Comput. Chem.*, 53:209–214, 2005.

[67] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17(3):449–467, 1965.

[68] P. Erd˝ os. Graph theory and probability. *Canadian Journal of Mathematics*, 11(1):34, 1959.

[69] P. Erd˝ os and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. *Infinite and finite sets*, 2:609–627, 1975.

[70] P. Erd˝ os, A.L. Rubin, and H. Taylor. Choosability in graphs. *Congr. Numer*, 26:125–157, 1979.

[71] P. Erdös and A. Hajnal. Ramsey-type theorems* 1. *Discrete Applied Mathematics*, 25(1-2):37–52, 1989.

[72] P. Erdös and A.H. Stone. On the structure of linear graphs. *Bull. Amer. Math. Soc*, 52:1087–1091, 1946.

[73] L. Esperet, M. Montassier, and A. Raspaud. Linear choosability of graphs. *Discrete Mathematics*, 308(17):3938–3950, 2008.

[74] L. Esperet, M. Montassier, and A. Raspaud. Linear choosability of graphs. *Discrete Mathematics*, 308(17):3938–3950, 2008.

[75] R. Fagin. Degrees of acyclicity for hypergraphs and relational database schemes. *Journal of the ACM (JACM)*, 30(3):514–550, 1983.

[76] A.M. Farley and A. Proskurowski. Extremal graphs with no disconnecting matching. *Congressus Nummerantium*, 41:153–165, 1984.

[77] Guillaume Fertin and André Raspaud. Acyclic coloring of graphs of maximum degree Δ. In Felsner Stefan, editor, *2005 European Conference on Combinatorics, Graph Theory and Applications (EuroComb '05)*, volume AE of *DMTCS Proceedings*, pages 389–396. Discrete Mathematics and Theoretical Computer Science, 2005.

[78] I. Fiamčik. The acyclic chromatic class of a graph (Russian). *Math. Slovaca*, 28(2):139–145, 1978.

[79] A. Fiedorowicz, M. Haluszczak, and N. Narayanan. About acyclic edge colourings of planar graphs. *Information Processing Letters*, 108(6):412–417, 2008.

[80] F.V. Fomin and D. Kratsch. *Exact exponential algorithms*. Springer Verlag, 2010.

[81] P. Frankl. Extremal set systems. In *Handbook of combinatorics (vol. 2)*, pages 1293–1329. MIT Press, 1996.

[82] P. Frankl and V. Rödl. Near perfect coverings in graphs and hypergraphs. *Europ. J. Combinatorics*, 6:317–326, 1985.

[83] T. Gallai. *On directed paths and circuits*. Academic Press, New York, 1968.

[84] F. Galvin. The list chromatic index of a bipartite multigraph. *Journal of Combinatorial Theory, Series B*, 63(1):153–158, 1995.

[85] Glpk – gnu linear programming kit. `http://www.gnu.org/software/glpk/`.

[86] C.D. Godsil, G. Royle, and CD Godsil. *Algebraic graph theory*, volume 8. Springer New York, 2001.

[87] D.A. Grable. More-than-nearly-perfect packings and partial designs. 1994.

[88] M.H. Graham. On the universal relation. *University of Toronto technical report*, 1979.

[89] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[90] I. Gutman and N. Trinajstic. Graph theory and molecular orbitals. Total [phi]-electron energy of alternant hydrocarbons. *Chemical Physics Letters*, 17(4):535–538, 1972.

[91] S. Gutner and M. Tarsi. Some results on (a: b)-choosability. *Discrete Mathematics*, 309(8):2260–2270, 2009.

[92] A. Gyárfás. On ramsey covering numbers. *Infinite and Finite Sets*, 2:801–816, 1975.

[93] A. Gyárfás, E. Szemerédi, and Z. Tuza. Induced subtrees in graphs of large chromatic number. *Discrete Mathematics*, 30(3):235–244, 1980.

[94] R. Häggkvist and A. Chetwynd. Some upper bounds on the total and list chromatic numbers of multigraphs. *Journal of graph theory*, 16(5):503–516, 1992.

[95] S.G. Hartke and AJ Radcliffe. Mckay's canonical graph labeling algorithm. In *Communicating mathematics: a conference in honor of Joseph A. Gallian's 65th birthday, July 16-19, 2007, University of Minnesota, Duluth, Minnesota*, volume 479, page 99. Amer Mathematical Society, 2009.

[96] Alan Hartman. Software and hardware testing using combinatorial covering suites. In Martin Charles Golumbic and Irith Ben-Arroyo Hartman, editors, *Graph Theory, Combinatorics and Algorithms*, volume 34 of *Operations Research/Computer Science Interfaces Series*, pages 237–266. Springer US, 2005.

[97] M. Hasse. Zur algebraischen begründung der graphentheorie. i. *Mathematische Nachrichten*, 28(5-6):275–290, 1965.

[98] F. Havet and L. Sampaio. On the grundy number of a graph. In *Proceedings of the International Symposium on Parameterized and Exact Computation(IPEC)*, number 6478 in Lecture Notes on Computer science, pages 170–179, December 2010.

[99] S. Hedetniemi and P. Slater. Line graphs of triangleless graphs and iterated clique graphs. *Graph theory and Applications*, pages 139–147, 1972.

[100] Luc Hogie, Aurélien Lancin, Nathann Cohen, and Issam Tahiri. Grph – a java library for the manipulation of graphs. `http://www-sop.inria.fr/members/Luc.Hogie/grph/`.

[101] Ken ichi Kawarabayashi, Yusuke Kobayashi, and Bruce Reed. The disjoint paths problem in quadratic time. *Journal of Combinatorial Theory, Series B*, In Press, Corrected Proof:–, 2011.

[102] A. Iranmanesh, I. Gutman, O. Khormali, and A. Mahmiani. The edge versions of the wiener index. *MATCH Commun. Math. Comput. Chem.*, 61:663–672, 2009.

[103] Gutman Ivan and Estrada Ernesto. Topological indices based on the line graph of the molecular graph. *J. Chem. Inf. and Comput. Sci.*, 36(3):541–543, 1996.

[104] TR Jensen and B. Toft. *Graph Coloring Problems*. 1995.

[105] Hou Jianfeng, Liu Guizhen, and Wang Guanghui. Improved bounds for acyclic chromatic index of planar graphs. *Discrete Applied Mathematics*, In Press, Corrected Proof:–, 2011.

[106] Lawrence Jim, N.Kacker Raghu, Lei Yu, Kuhn D.Richard, and Forbes Michael. A Survey of Binary Covering Arrays. *Electronic Journal of Combinatorics*, 18(1), 2011.

[107] S. Jukna. *Extremal combinatorics: with applications in computer science*. Springer Verlag, 2001.

[108] J. Kahn. Asymptotics of the list-chromatic index for multigraphs. *Random Structures & Algorithms*, 17(2):117–156, 2000.

[109] H.A. Kierstead and S.G. Penrice. Radius two trees specify χ-bounded classes. *Journal of Graph Theory*, 18(2):119–129, 1994.

[110] A. Kotzig. Hamilton graphs and Hamilton circuits. In *Theory of Graphs and its Applications, Proceedings of the Symposium held in Smolenice in June*, pages 63–82, 1963.

[111] K. Kuratowski. Sur le probleme des courbes gauches en topologie. *Fund. Math*, 15(271-283):79, 1930.

[112] Jean-François Lalande, Michel Syska, and Yann Verhoeven. Mascopt - A Network Optimization Library: Graph Manipulation. Technical Report RT-0293, INRIA, 2004.

[113] L. Lovász. Kneser's conjecture, chromatic number, and homotopy. *Journal of Combinatorial Theory, Series A*, 25(3):319–324, 1978.

[114] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM (JACM)*, 41(5):960–981, 1994.

[115] R.C.S. Machadoa, C.M.H. de Figueiredob, and N. Trotignon. Edge-colouring and total-colouring chordless graphs. *Submitted to Theoretical Computer Science*, 2010.

[116] J. Matousek. *Using the Borsuk-Ulam Theorem*. Springer, 2003.

[117] B.D. McKay. Nauty user's guide (version 1.5). *Computer Science Department, Australian National University, GPO Box*, 4.

[118] B.D. McKay and Vanderbilt University. Dept. of Computer Science. *Practical graph isomorphism*. Citeseer, 1981.

[119] B. Mohar and C. Thomassen. Graphs on surfaces. johns hopkins studies in the mathematical sciences, 2001.

[120] R.A. Moser and G. Tardos. A constructive proof of the general Lovász Local Lemma. *Journal of the ACM (JACM)*, 57(2):1–15, 2010.

[121] Z. Tuza N. Cohen. Induced decompositions of graphs (manuscript).

[122] A. Nakayama and B. Peroche. Linear arboricity of digraphs. *Networks*, 17(1):39–53, 1987.

[123] J. Naor and M. Naor. Small-bias probability spaces: efficient constructions and applications. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 213–223. ACM, 1990.

[124] C. Nash-Williams. Decomposition of finite graphs into forests. *Journal of the London Mathematical Society*, 1(1):12, 1964.

[125] Networkx – high productivity software for complex networks. `http://networkx.lanl.gov/`.

[126] S. NIKOLIC, G. KOVACEVIC, A. Milicevic, and N. TRINAJSTIC. The zagreb indices 30 years after. *Croatica chemica acta*, 76(2):113–124, 2003.

[127] Sampo Niskanen and Patric R. J. Östergård. Cliquer - routines for clique searching, 2002. `http://users.tkk.fi/pat/cliquer.html`.

[128] N. Nisse. *Jeux des gendarmes et du voleur dans les graphes. Mineurs de graphes, stratégies connexes, et approche distribuée.* PhD thesis, Université Paris Sud, 2007.

[129] J.P.C. Peterson. Die theorie der regulären graphen. *Acta Mathematica*, 15:193–220, 1891.

[130] T. Pitassi and A. Urquhart. The complexity of the hajos calculus. 1992.

[131] M. Randic and N. Trinajstic. In search for graph invariants of chemical interest. *Journal of molecular structure*, 300:551–571, 1993.

[132] A. Raspaud and W. Wang. Linear colouring of planar graphs, 2006. Research report `http://www.labri.fr/perso/lepine/Rapports_internes/RR-138406.ps.gz`.

[133] A. Raspaud and W. Wang. Linear coloring of planar graphs with large girth. *Discrete Mathematics*, 309(18):5678–5686, 2009.

[134] D.K. Ray-Chaudhuri and R.M. Wilson. Solution of kirkman's schoolgirl problem. In *Proc. Symp. Pure Math*, volume 19, pages 187–203, 1971.

[135] M. Richardson. On weakly ordered systems. *Bull. Amer. Math. Soc*, 52:113–116, 1946.

[136] N. Robertson, D. Sanders, P. Seymour, and R. Thomas. The four-colour theorem* 1. *Journal of Combinatorial Theory, Series B*, 70(1):2–44, 1997.

[137] N. Robertson and P.D. Seymour. Graph minors series (I-XXIII. *Journal of Combinatorial Theory, Series B*, 1983-2011.

[138] B. Roy. Nombre chromatique et plus longs chemins d'un graphe. *Revue française d'informatique et recherche opérationnelle*, 1(5):129–132, 1967.

[139] D.P. Sanders and Y. Zhao. Planar graphs of maximum degree seven are class I. *Journal of Combinatorial Theory, Series B*, 83(2):201–212, 2001.

[140] E.R. Scheinerman and D.H. Ullman. *Fractional graph theory*. New York: John Wiley and Sons, 1997.

[141] J.P. Schmidt and A. Siegel. The Spatial Complexity of Oblivious k-Probe Hash Functions. *SIAM Journal on Computing*, 19:775, 1990.

[142] W. Schnyder. Planar graphs and poset dimension. *Order*, 5(4):323–343, 1989.

[143] A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*. Springer Verlag, 2003.

[144] Stefan Behnel, Robert Bradshaw, Dag Sverre Seljebotn, Greg Ewing, et al. Cython: C-Extensions for Python, 2011.

[145] W. A. Stein et al. Sage Mathematics Software (Version 4.7), 2011. `http://www.sagemath.org`.

[146] D.P. Sumner. Subtrees of a graph and chromatic number. *The Theory and Applications of Graphs, G. Chartrand (ed.), John Wiley, New York*, 1981.

[147] C. Thomassen. Every planar graph is 5-choosable. *Journal of Combinatorial Theory, Series B*, 62(1):180–181, 1994.

[148] Helge Tverberg. On a lovász-type lemma, applied to brooks' theorem for list-colouring. *Discrete Mathematics*, 311(17):1995 – 1997, 2011.

[149] J. Van Den Heuvel and S. McGuinness. Coloring the square of a planar graph. *Journal of Graph Theory*, 42(2):110–124, 2003.

[150] Guido van Rossum. Python, 2011.

[151] LM Vitaver. Determination of minimal coloring of vertices of a graph by means of boolean powers of the incidence matrix (russian). In *Dokl. Akad. Nauk SSSR*, volume 147, pages 758–759, 1962.

[152] V.G. Vizing. On an estimate of the chromatic class of a p-graph. *Diskret. Analiz*, 3(7):25–30, 1964.

[153] VG Vizing. Critical graphs with given chromatic class. *Diskret. Analiz*, 5:6–17, 1965.

[154] M. Voigt. List colourings of planar graphs. *Discrete Mathematics*, 120(1-3):215–219, 1993.

[155] D Vukičević and A Graovac. Comparing zagreb m_1 and m_2 indices for acyclic molecules. *MATCH Commun. Math. Comput. Chem.*, 57(3):587–590, 2007.

[156] Jianfang Wang. *The information hypergraph theory*. Science Press, Beijing, 2008.

[157] D. Yu, J. Hou, G. Liu, B. Liu, and L. Xu. Acyclic edge coloring of planar graphs with large girth. *Theoretical Computer Science*, 410(47-49):5196–5200, 2009.

[158] X. Zhu. On-line list colouring of graphs. *Electronic Journal of Combinatorics*, 16(R127):1, 2009.

[159] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(1):103–128, 2007.